THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Efficient Development of Vehicle Configuration Rules

ANNA TIDSTAM

Department of Product and Production Development

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2014

Efficient Development of Vehicle Configuration Rules
ANNA TIDSTAM
ISBN 978-91-7385-963-9

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr. 3644
ISSN 0346-718X

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 shows the vehicle configuration process from a customer perspective. The vehicle configuration process is based on three classes of vehicle configuration rules, the development of which is the topic of this PhD thesis.

Printed by
Chalmers Reproservice
Gothenburg, Sweden, 2014

Abstract

The aim of this PhD thesis is to describe an efficient and user-friendly development process for vehicle configuration rules. The development of vehicle configuration rules precedes the sales configuration, and is the process of authoring, evaluating and releasing configuration rules sets. The vehicle configuration rules are logical expressions using IF, AND, NOT, OR etc. An example of a vehicle configuration rule is IF(*seat heating ventilation*) THEN(*comfort seats*).

The research method used was analysing the development of vehicle configuration rules by focusing on 1) configuration information models, 2) configuration rules visualisation methods and 3) configuration rules development methods. The result from the first research paper was a vehicle configuration information model, on which all the following research papers were subsequently based. The configuration information model should therefore be viewed as this PhD thesis' most central research contribution. The aim of a more efficient and user-friendly development process for vehicle configuration rules was fulfilled by developing a new configuration rules visualisation method. This new matrix-based configuration rules visualisation method uses an inference engine. The inference engine draws conclusions based on computations of the vehicle configuration rules. The conclusions are then shown in the matrix-based vehicle configuration rules visualisation. The computations were previously done manually by the product developer, and hence some training and time were required to undertake the configuration rules analysis.

The usability and time-efficiency were evaluated with formative usability tests. These tests were conducted with product developers from the automotive industry. The product developers answered the test questions within minutes instead of hours/days/weeks.

The conclusion is that development processes for vehicle configuration rules could become more time-efficient and user-friendly by developing the configuration rule visualisation tools. The introduction of inference engines addresses the needs of the product developers. More research is, however, needed on the development of vehicle configuration rules, in particular, research is needed on complexity management and product rationalisation.

Keywords: vehicle configuration; configuration rules; development process; matrix-based visualisation; information modelling; process modelling.

Acknowledgements

To start with, I would like to thank my supervisor Professor Johan Malmqvist for his strategic feedback and positive attitude. I also thank my co-supervisor Hans Johannesson for critically questioning my research approach. Thanks also to my co-authors Alexey Voronov, Knut Åkesson and Martin Fabian from the division of Automation at Chalmers, Lars-Ola Bligård from the division of Design and Human Factors at Chalmers and Fredrik Ekstedt from the Fraunhofer-Chalmers Research Centre for Industrial Mathematics.

Numerous people have contributed from the automotive industry. Without you it would not have been possible to write this thesis. Some of you have gone far beyond what could reasonably have been requested from official work duties. I will be forever grateful for your support.

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. An international research visit was financed by ProViking. The financial support is gratefully acknowledged.

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

Anna Tidstam,
Gothenburg, 2014

Appended Papers

The appended Papers A, B, C, D, E, F, and G have been written for this PhD thesis. Each paper is presented here with a brief description of 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.

Johan Malmqvist and Anna Tidstam planned the paper. Anna Tidstam collected the empirical data at a vehicle manufacturing company. Anna Tidstam analysed 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, pp. 33-46.

The paper was planned by Anna Tidstam and Johan Malmqvist. Anna Tidstam and an employee from one of the companies studied collected the empirical data through user studies. Anna Tidstam analysed the data and wrote the paper. 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, pp. 305-318.

Anna Tidstam and Johan Malmqvist planned the paper. Anna Tidstam collected the CR visualisation requirements together with one employee from a vehicle manufacturing company that was studied. 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 one of the vehicle manufacturing companies studied conducted the usability tests. The paper was written by Anna Tidstam. All co-authors reviewed the paper.

Paper D Tidstam, A. & Malmqvist, J. (2013), "Obstacles and Development of Support for Translation of Configuration Rules", *Proceedings of the International Conference on Engineering Design (ICED13),* Seoul, Korea, 19-22 July, pp. 119-128.

Johan Malmqvist and Anna Tidstam planned the paper. Anna Tidstam studied the exchange process during a 3-month stay at a vehicle manufacturing firm. Anna Tidstam coded the algorithm. The paper was written by Anna Tidstam and reviewed by Johan Malmqvist.

Paper E Tidstam, A. & Malmqvist, J. (2013), "Comparison of Configuration Rule Visualization Methods", *Proceedings of Product Lifecycle Management (PLM13),* Nantes, France, 6-10 July, pp. 550-559.

Anna Tidstam studied the configuration rules visualisation methods during a 3-month stay at one of two collaborating vehicle manufacturing firms. Johan Malmqvist and Anna Tidstam planned the paper. The paper was written by Anna Tidstam and reviewed by Johan Malmqvist.

Paper F Tidstam, A. & Malmqvist, J. (2013), "A Systematic Process for Developing Vehicle Configuration Rules", *Submitted to Scientific Journal.*

The paper was planned by Anna Tidstam and Johan Malmqvist. Anna Tidstam conducted the literature review and modelled the development process for vehicle configuration rules. The paper was written by Anna Tidstam and reviewed by Johan Malmqvist.

Paper G Tidstam, A., Voronov, A., Åkesson, K., Malmqvist, J. & Fabian, M. (2014), "Formal methods for the inspection of vehicle configuration rules", *Submitted to Scientific Journal.*

Anna Tidstam, Alexey Voronov, Knut Åkesson and Martin Fabian planned the paper. Alexey Voronov, Knut Åkesson and Martin Fabian formalised the computer support. Anna Tidstam wrote the paper, with the exception of Alexey Voronov's description of the formal methods and explanations of the computation feasibility. The paper was reviewed by all co-authors.

Acronyms

AB .... Aktiebolag (English: corporation)
AG .... Aktiengesellschaft (English: corporation)
AP214 .... Application protocol 214
ARC .... Area of relevance and contribution
BMW .... Bayerische Motoren Werke (English: Bavarian Motor Works)
BOM .... Bill-of-Material
CAD .... Computer-aided design
CM .... Configuration management
CR .... Configuration rule
CRI .... Configuration request interfaces
CSP .... Constraint satisfaction problem
DRM .... Design research methodology
DSM .... Design structure matrix
GPDS .... Global Product Development System
IDEF .... Integration definition for function modelling
IEC .... International electro-technical commission
ISO .... International organisation for standardisation
IT .... Information technology
KBE .... Knowledge-based engineering
KBS .... Knowledge-based system
KDP .... Konstruktion Data Personvagnar (English: engineering data for cars)
KOLA .... Konstruktionsdata Lastvagnar (English: engineering cata for trucks)
OMG .... Object Management Group
MDM .... Multiple-domain matrix
MIKE .... Model-based and Incremental Knowledge Engineering

MOKA...........Methodology and tools Oriented to Knowledge-based engineering Applications
PDM ....Product data management
PFMP............Product family master plan
PLM ....Product lifecycle management
PMR..............Product modification request
PCS ....Product configuration system
PDMA ....Product Development and Management Association
PTC ....Parametric Technology Corporation
RG.................Research goal
RHS...............Right hand side
RQ ....Research question
SCM ....Software configuration management
STEP.............Standard for the exchange of product data
UML ....Unified modelling language
VDA ....Verband der Automobilindustrie (English: German Association of the Automotive Industry)
VPI ....Variant parameter interface

Table of Contents

1 Introduction

This introductory chapter describes the background, defines the problem, states the purposes and goals, describes the scope and finally outlines the remainder of the PhD thesis.

1.1 Background

There are many examples of configurable products, e.g. computers, networks, buildings, bikes and trucks. A commonly cited definition of configuration is:

Configuration is a special case of design activity where the artefact 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 mentions that the components can only be connected in certain ways. This knowledge is stored in configuration rules. This section gives an introduction to vehicle configuration by addressing the *vehicle configuration information model*, the *vehicle configuration rule development process* and *product configuration systems.*

1.1.1 Vehicle configuration information model

A configurable product has a set of concepts that are used when formulating configuration rules. The concepts are described formally by an *information model*. An information model is a representation of concepts, relationships, constraints, rules and operations to specify data semantics (Lee, 1999). This thesis studies vehicle configuration, and a configured vehicle is usually specified by a list of *feature variants*, e.g. *exterior colour white* or *exterior colour red* (Tidstam & Malmqvist, 2010). There can be hundreds of feature families/variants for vehicles, e.g. engine sizes, exterior colours, interior equipment, wheel suspensions etc. This means that a large number of unique vehicle configurations are possible. Feature variants do not necessarily correspond to specific physical vehicle components; they could also define functionalities offered to the customers. Vehicle functionality may be offered by feature variants for certain road conditions, e.g. smooth or rough roads. As shown in Table 1.1, it is common that feature variants have codes, e.g. RC-ROUGH and RFUEL490. All feature variants belong to *feature families*, which are groupings of similar feature variants. Most feature families also have a condition that exactly one feature variant from the feature family has to be included in each product configuration. For example, feature family *RHS fuel tank* in Table 1.1 allows exactly one RHS fuel tank, e.g. the feature variant *490 litres*, in a truck configuration.

Table 1.1: Examples of truck feature variants (Lindroth, 2011).

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

The vehicle configuration is an example of two-level configuration (Haag, 1998). The two steps are called *high-level* and *low-level* configuration. High-level configuration is the selection of feature variants. An example of a configuration rule for high-level configuration is *NOT(17 inch wheel AND 20 inch tyre)*. Both restricting and allowing relationships are called configuration rules (Euwe & Schuwer, 1993). Low-level configuration follows from the high-level configuration, and implies a set of *items*. Items are components, interfaces, drawings or other documents required to manufacture the vehicle configuration. The development of vehicle configuration rules precedes the high- and low-level configuration.

1.1.2 Vehicle configuration rules development process

As was stated in (Haag, 1998; Sabin & Weigel, 1998), it is useful to distinguish between sales configurators and configuration for manufacturing and engineering. Sales configurators are business-to-customer applications, with much less detail than is needed for manufacturing. The development of CR precedes the sales to delivery process, as shown in Fig. 1.1. The input to the development of vehicle CR is a product modification request, which may require an addition, modification or

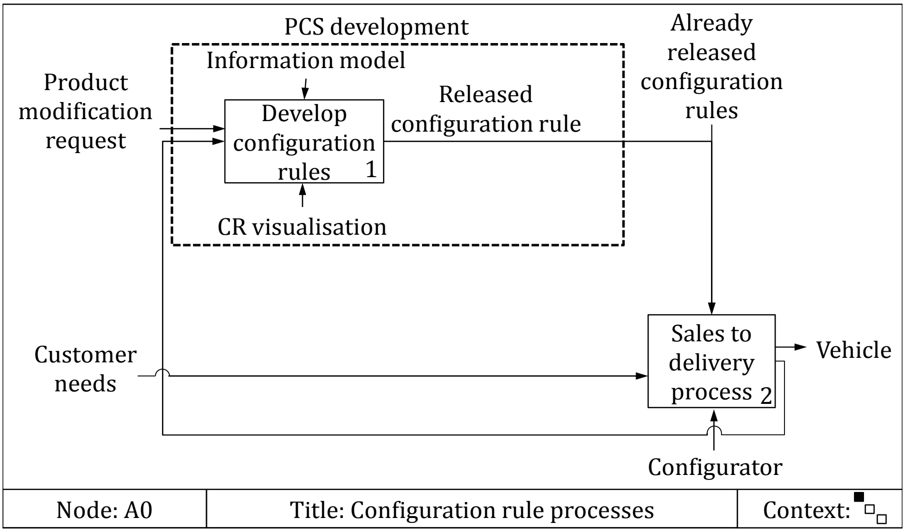


Figure 1.1: Process model for the development and use of vehicle CR.

deletion of a CR. The output from the CR development process is a released CR fulfilling the product modification request.

There are variations of the CR development process. The CR development process may take place during:

1. The traditional new development process at a vehicle manufacturing company;
2. The traditional new development process at a vehicle manufacturing company acting as a supplier;
3. The product rationalisation process at a vehicle manufacturing company.

Further details about the CR development process will be described in the Frame of Reference section. The level of details presented here is sufficient for describing the problem analysis, which is the topic of the next section.

Fig. 1.1 also specifies the development of product configuration systems (PCS), the topic of the next section.

1.1.3 Product configuration systems

Traditional computer systems cannot understand the information stored in them. This is in contrast with knowledge-based systems (KBS), which have an inference engine that derives conclusions from the information (Akerkar & Sajja, 2010). There are always two elements in KBS systems, the inference engine and the *knowledge base*. The knowledge base is information expressed using some formal knowledge representation language (Foldoc, 1994). There are several KBS formalisms, such as *rules*, *logic*, *models* and *cases* (Sabin & Wiegel, 1998). Following KBS formalism definitions, rule-based KBS contain IF-THEN actions that simulate human problem solving. Rule-based KBS are also called *expert systems* due to their similarity to human experts.

If KBS systems are applied to on product configuration, the systems are called *product configuration systems*. A knowledge base in product configuration systems can be expressed using any of the KBS formalisms. Many different formalisms have been applied because of the differences in the products to be configured (Junker, 2006). The knowledge base must, however, represent a configurable product (Edwards & Lindgaard Pedersen, 2004). The product configuration system configures products by executing the inference engine in relation to the knowledge.

Heinrich and Jüngst (1991) have studied how knowledge is used for configuration. They found four categories of knowledge: system knowledge, catalogue knowledge, evaluation knowledge and performance knowledge:

- *System knowledge* is defined as the relatively stable product modularisation;
- *Catalogue knowledge* populates the product modularisation with platform-specific elements, e.g. feature variants and items;
- *Evaluation knowledge* is a property evaluated for a configuration, e.g. sales merit information;
- *Performance knowledge* aids the configuration process so that it as efficient as possible.

This thesis studies the development of CRs, and hence the development of catalogue knowledge. Evaluation knowledge will also be further examined in terms of the product developers' evaluation knowledge of which configurations should be allowed. This evaluation knowledge is currently not encoded into CR, but is a knowledge that the product developers have.

The next section will describe the CR development process, as well as the two configuration use processes: sales configuration and configuration of bill-of-materials.

1.2 Problem analysis

Researchers have found that it is not humanly possible to keep the set of vehicle CR absolutely defect-free (Sinz et al., 2003). The objective is to minimise the errors, as well as to make the development process more time-efficient. Examples of errors are configurations permitted by the configurator that cannot be manufactured. Examples of reasons why they cannot be manufactured are insufficient space in which to fit the components, or components that are missing. Another error is faulty or missing CR making the sale configurator reject vehicle configurations that should be permitted. Rejected vehicle configurations that should have been accepted result in loss of income.

The following sections will describe the problems related to the CR information model, the CR development process and the CR visualisation methods.

1.2.1 CR information model

There are several suggestions for how to model product structures and associations, e.g. the *generic BOM* (Van Veen, 1992), *high-level* and *low-level* configuration (Haag, 1998) and the information model standard *AP214* (ISO, 2004). Another approach to model configuration rules is when formulating a Constraint Satisfaction Problem (CSP). The benefit of formulating a CSP is that algorithms exist for how to compute the configurations. The CSP only consists of variables with corresponding domains as well as constraints. The problem is that it is difficult to express vehicle configuration rules as a CSP. For example, there is huge level of details in the AP214 compared with a CSP.

The next section discusses the CR development process and identifies two of its time-consuming activities.

1.2.2 CR development process

The three variations of CR development processes have their own specific problems. Each variation will now be discussed in terms of the problems identified in relation to it:

1. **The traditional new development process at an automotive manufacturing company.** A problem often cited as occurring during the CR development process is the *knowledge acquisition bottleneck* (Schreiber et al., 1993), i.e. the iterations between product structure specialists and design engineers. Schreiber et al conclude that it is very difficult to extract and formalise the necessary information from the people who possess the knowledge.
2. **The traditional new development process at an automotive manufacturing company acting as a supplier.** Besides knowledge acquisition bottlenecks, an exchange of configuration rules also needs to take place between the two companies. Exchange of product data has recognised problems, e.g. incompleteness and time-delays (Domazet et al., 2000).
3. **The product rationalisation process at an automotive manufacturing company.** Vehicles are becoming increasingly complex. Each product modification tends to erode the system built to manage product complexity (Jacobson & Lindström, 1991).

Another reason why it is argued that CR development processes are error-prone and time-consuming is that they rely in part on manual inspections (Baumeister & Freiberg, 2010). The manual inspection is undertaken with various CR visualisation methods, which is the topic of the next section.

1.2.3 CR visualisation

One problem with the existing vehicle CR visualisation methods is that they are not easy to use. Infrequent users at the truck manufacturer Scania describe the company's CR visualisation tool as inaccessible (Pak, 2011). Knowledge about how the tool works and what it can be used for is not widely disseminated. What happens is that instead of learning how to use it, the less frequent users ask the more experienced users to provide the information they want. The consequence is that more people are involved than necessary, and it is more difficult to determine which data is out-of-date.

An example of a user interface from an industrial CR visualisation tool for vehicle configuration rules is shown in Fig. 1.2. The long text string starting with "–

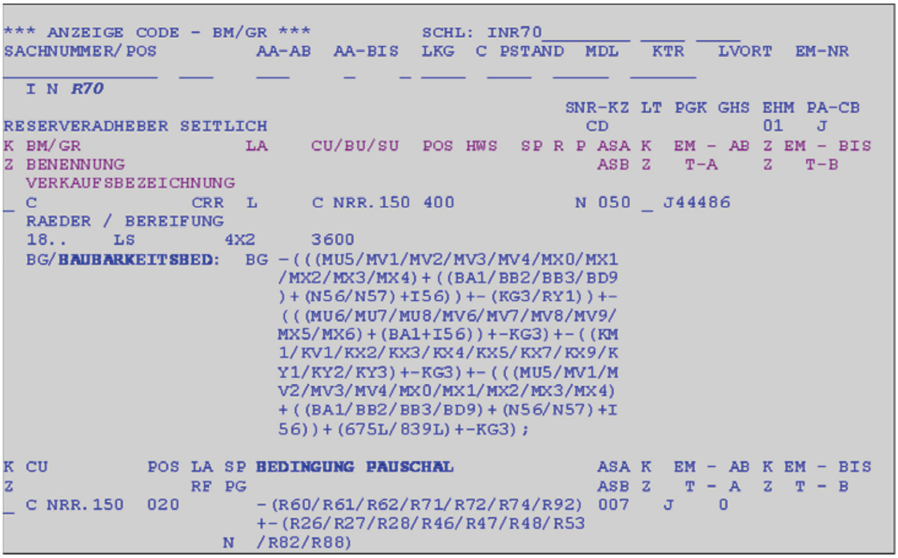


Figure 1.2: Text-based CR visualisation, here using only feature variants (Hami-Nobari & Blessing, 2005).

((MU5/…" is a configuration rule including 63 feature variants. Each of the codes are 3-4 characters long, corresponding 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."

An easy-to-use CR visualisation tool is particularly needed for less frequent users, e.g. design engineers who are able to validate the configuration rules but are also occupied with many other duties. Product structure specialists are today often required to support design engineers in analysing the configuration rules.

The problems identified are the justification for the thesis' purpose, as described in the next section.

1.3 Purpose of Thesis

The general purpose of this PhD thesis is to make the CR development process more efficient and less error-prone. The hypothesis is that facilitating the CR development process decreases the number of errors discovered at the assembly line. The challenge 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. How to accomplish the purpose of the thesis is formulated using the research goal in the following section.

1.4 Goal of Thesis

The aim of a more efficient and less error-prone vehicle configuration rules process is addressed with a research goal (RG):

RG: To create a systematic and efficient development process for vehicle configuration rules.

Based on this overall goal and an analysis of the literature in the field, a number of research needs were identified (Section 2.6) and research questions formulated (Section 3.1). The literature is discussed in Chapter 2 with a review of applicability to the context of this PhD thesis. The research questions are discussed further in Chapter 3 in order to show how they relate to existing literature.

1.5 Scope of Thesis

The information modelling conducted for this PhD thesis has some delimitations. This thesis has no intention of extending the lifecycle support, e.g. by supporting early phases of product development with a design rationale. Another limitation on scope is that downstream information systems should not be affected, as this would hinder implementation of the research results. Some researchers within the area of configuration argue that it is the number of configuration rules that is making it time-consuming to develop them (Hami-Nobari & Blessing, 2005; Huang et al., 2008). These authors prescribe a more sophisticated information modelling language for the vehicle configuration rules, e.g. by including relational expressions (*less than*, *more than*) for the configuration rules, which would affect downstream systems. The scope of this PhD thesis is to remain with the industrial vehicle configuration rules, which are limited to logic expressions (NOT, OR, AND, IF-THEN etc.).

Another limitation is that the integration of software configuration management, SCM, and PDM has not been studied. Software is considered a special type of item, with configuration rules that require analysis using SCM.

1.6 Outline of Thesis

The remainder of this PhD thesis is organised as follows. Chapter 2 describes the frame of reference and identifies the research needs. Chapter 3 describes the research method including discussion of the more specific research questions that were derived from the research needs identified. Chapter 4 summarises 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

The frame of reference is described using a system-based framework. Firstly, there is a discussion concerning the framework, as well as other system perspectives on development of configuration rules. Then, literature is reviewed by referring to the elements in the framework. Finally, the research needs are identified and analysed.

2.1 Perspectives on configuration

The development of configuration rules may be studied from several perspectives. A review of the literature concerning system frameworks for engineering information management systems by (Burr et al., 2007) found six system layers: *processes* as the top-level system element, followed by *information systems*, *methods*, *information*, *data* and *organisation* as the five remaining layers. This system hierarchy reflects the fact that a process model, for example IDEF0 models, incorporates all other system elements.

Turesson (2006) describes another framework consisting of the system elements: *process*, *methods*, *information*, *applications* and *infrastructure*. Turesson's framework is called *Pater-Noster*. Zimmerman (2008) proposes a system framework where the information, application and infrastructure are motivated by business processes, business strategy and business objectives.

The framework selected for describing the research work in this PhD thesis has the system elements: *processes*, *information*, *information systems* and *organisation*, see Fig. 2.1 (Svensson et al., 1999). The information system includes both the application and infrastructure with the Pater-Noster approach.

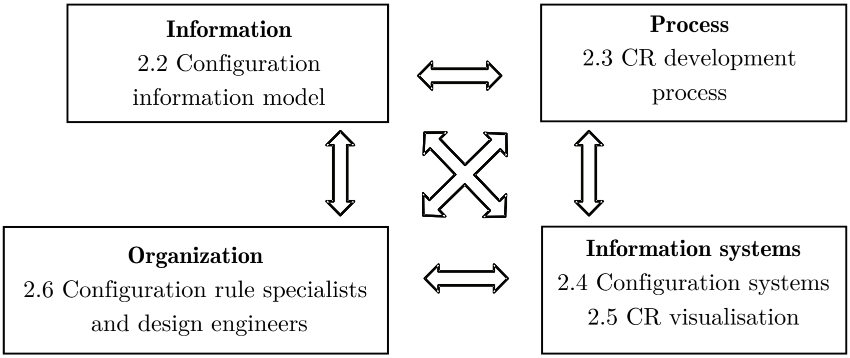


Figure 2.1: System-based framework for describing engineering information systems, adapted from (Svensson et al., 1999).

The arrows in the figure indicate that all system elements contribute to the system's purpose, and none of the parts can be approached as separate phenomena, as stated by Hitchins (2003). Note that each system element will be addressed in a separate chapter (2.2-2.6), as also shown in the figure.

2.2 CR information models

This section starts with definitions of product structures and product models, before in separate sections reviewing the literature addressing the two terms.

2.2.1 Product structures and product modelling

Product structure is defined as a set of elements and their interrelationships that describe how a product is built up from a particular viewpoint (Svensson & Malmqvist, 2002). Traditionally, the vehicle product structure mainly contained physical parts, including all assemblies, subassemblies and parts for a product (Garwood, 1988). As vehicles include more and more software this is also reflected in the content of the product structure. A product structure for a single product variant is frequently called Bill of Material, henceforth BoM. Any modelling of a product is called *product modelling,* which can include the product structure modelling but also many more aspects, dimension or viewpoints of a product. The published literature on product models for configurable products is reviewed in the next section.

2.2.2 Product models for configurable products

Collier (2001) describes 12 product models in his *Twelve-Fold way*, see Fig. 2.2. The triangle in the Twelve-Fold way shows how any product model can be described as a transformation of two closely adjacent product models, e.g. the *order specifications* and the *manufacturing capabilities* give the *manufacturing process designs*. Multiple product models similar to the Twelve-Fold way are also discussed by Isaksson et al. (2000), e.g. product structure, product features, geometry representations, product configuration, manufacturing information and analysis information. Isaksson et al. do not create a complete product model but rather emphasise that a product model is not necessarily limited simply to the geometrical representation of the product. Another product model framework is called the *chromosome* product model, which contains four product models: product specification, function structure, organ structure and component structure (Andreasen, 1992). The chromosome model is based on the technical systems theory (Hubka & Eder, 1988), which states that these four different types of models are needed to describe a technical system and its development process. The chromosome model adds casual relationships between the four different product models to trace the origin of design characteristics, hence the word *chromosome*. The chromosome model is a parallel theory to the function-means tree. An example of a function-means tree is shown in Fig. 2.3. The *dry clothes* function is decomposed and determined by means such as *revolving drum*. The means are realised by parts. The function-means tree was first described by Tjalve

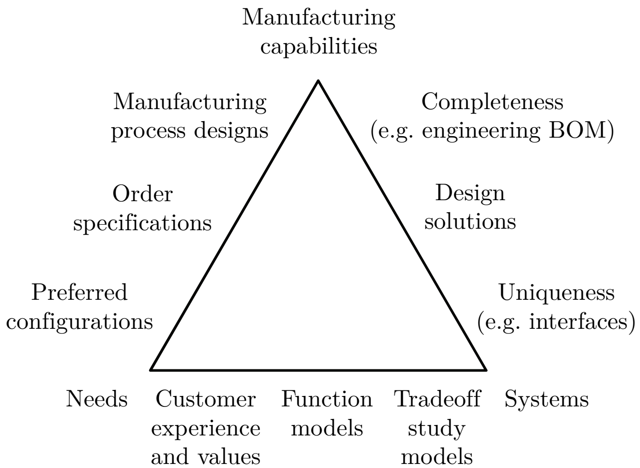


Figure 2.2: The Twelve-Fold way of product models (Collier, 2001).

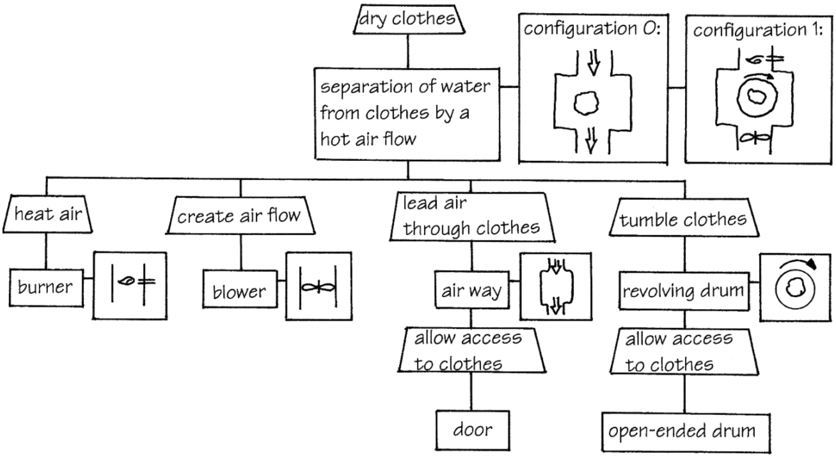


Figure 2.3: A function/means tree for a tumble dryer, adapted from (Hansen, 1995).

(1979), and is a graphical representation of Hubka's law which states that the developments of functions and parts are conducted in parallel. 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 thus have *causal links* stating the dependences between the domains.

The chromosome model, axiomatic design and function-means trees are typical examples of results deriving from research on the design process. However, they do not study how the configuration rules should be authored.

2.2.3 Product structures for configurable products

A framework for classifying product structures described in Claesson et al., (2001) includes the evolution from *closed* to *open* product structures. Claesson's framework has been slightly adapted in order to fit the categorisation of product structures used in this PhD thesis, see Table 2.1. Closed configuration information models represent one single product variant, e.g. a vehicle specification or a Bill of Material, while open configuration information models represent several product variants, i.e. a product family. Open product structures appeared in the early 1980s due to the increasing number of product variants that had to be managed. The term *feature variant* was introduced when the product variety was too large to define a product

Table 2.1: Framework for product structures and configuration processes.

	Product structures		
	Closed product structures	**Open (configurable) product structures**	
High level configuration	Vehicle specification	Generic BoM (Van Veen, 1992), Configurable components (Claesson, 2006)	Feature domain (product models, feature families, feature variants, product model authorisations, feature variant combination rules)
Low-level configuration	Bill of Material		Item domain (items and item usage rules)

number for each product configuration that was developed (Mather, 1982). The feature variants are used when describing variant parts. This is thus an example of a *two-level* configuration information model (Haag, 1998). The term was introduced as a characterisation of *high-level* configuration with feature variants and then *low-level* configuration for items.

What is important to emphasise for two-level product structures is that it is still possible to think of them as general configuration problems, i.e. constraint satisfaction problems (CSP) (Astesana et al., 2010a, Sinz et al., 2003). The CSP is defined as a triplet X, D and C, where X is a set of variables, D is a domain of values, and C is a set of constraints (Russell & Norvig, 2009). A solution to the CSP is an assignment of domain values to the variables that satisfies all constraints. The variables for high-level configuration are the feature families, with the domains corresponding to their feature variants. There are also variables for low-level configuration, which are the items with the domains either true or false depending on whether they are implied or not for a configuration. The constraints in the CSP are the configuration rules.

2.2.4 Configurable product structures

One example of a two-level configurable product structure is described by Mesihovic & Malmqvist (2004); see Fig. 2.4. High level configuration is specified by using feature variants, which are derived from module variety. For vehicles there are vehicle models, which play a crucial role in managing the complexity of vehicle product families. Each vehicle model, e.g. Volvo V70 or S40, has a specified module variety. The vehicle model is called product group in the figure. Low-level configuration is only executed when the high-level configuration has been completed. The blue lines indicate some relation between the objects which may be the configuration rules.

Another example of a two-level product structure is the generic Bill of Material, henceforth called the generic BOM (Van Veen, 1992). The generic BOM contains generic items, which are alternative (mutually exclusive) items. There are examples of generic items in the vehicle manufacturing industry, see Fig. 2.5. Here, the vehicle model is the top node of the item structure, which consists of several *main modules*. These are then divided into *modules*, which might also be further divided into *sub-*

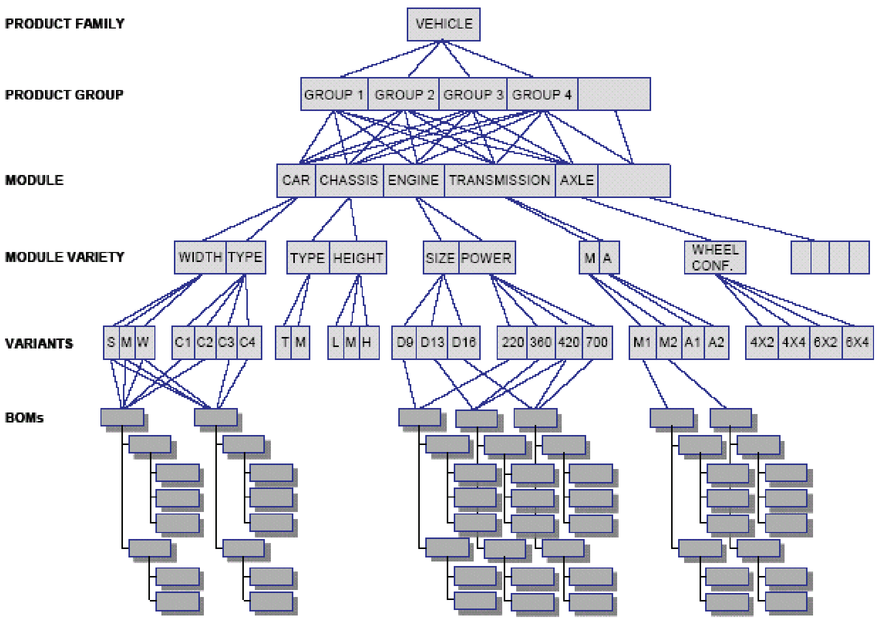


Figure 2.4: Example of two-level product structure (Mesihovic & Malmqvist, 2004).

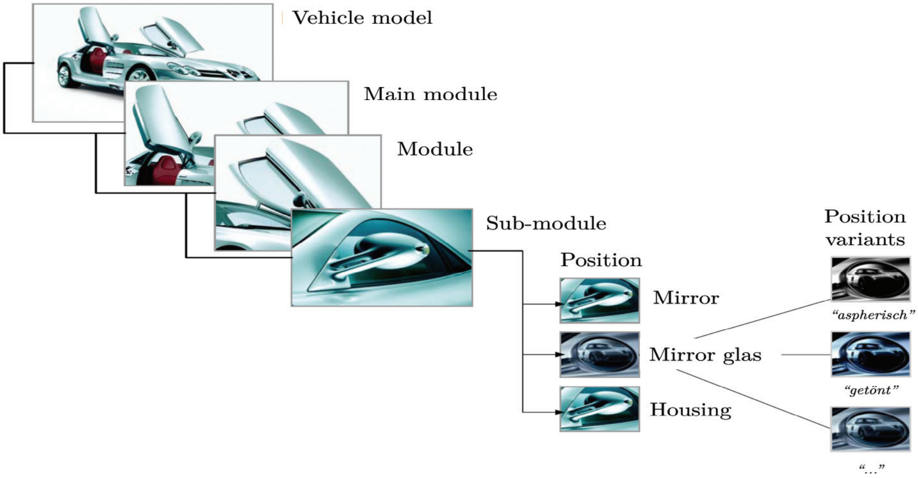


Figure 2.5: Generic items in terms of alternative mirror glass (Lamberti, 2011).

modules. In a sub-module, there are several *positions* which might have mutually exclusive items.

One of the application protocols for 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. 2.6. 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. AP214 is however a complex document with a large number of pages, and it is very difficult to get an overview of which classes are used to formulate the configuration problem. This is also the case for the comprehensive product structure described in Soininen et al. (1998). Additional elements *resources*, *connections* and *ports* have been introduced in the product structure as a synthesis of recognised configuration approaches. Soininen et al. (1998) argue that the suggested product structure is the most generic information model presented at the date of publication. The product structure indicates that it is not only logic-based configuration rules that are used, e.g. information element

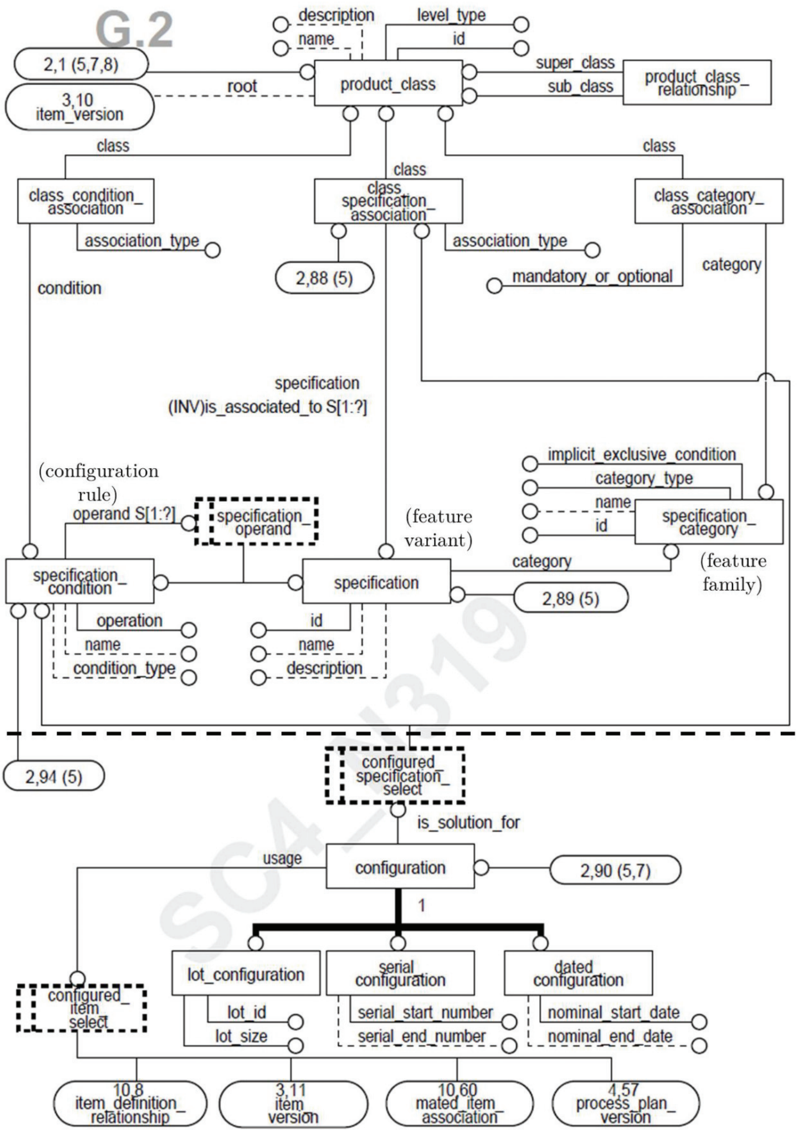


Figure 2.6: EXPRESS-G representations of objects used for the management of diversity in AP214 (ISO, 2004).

indicate the use of quantities.

The three examples shown of two-level product structures have been based on tree structures, i.e. there is a top node from where all the leaves can be reached. There is another variation of two-level product structures which are instead based on networks, so called *system-based* product structures. A system-based product structure is created by Collier (1999) through incorporating interface control objects between subsystems. These interface control objects then configure the product based on a specification of feature variants. The system-based product structure is also used in the configurable component approach (Claesson et al., 2001), see Fig. 2.7. The configurable component approach also uses interface control objects (called configuration request interfaces, CRI) as shown in the figure. The configurable component also has a parameter interface (VPI) with feature variants that can be used as input for the configuration rules inside the configurable component. The configurable components are autonomous, which means that all

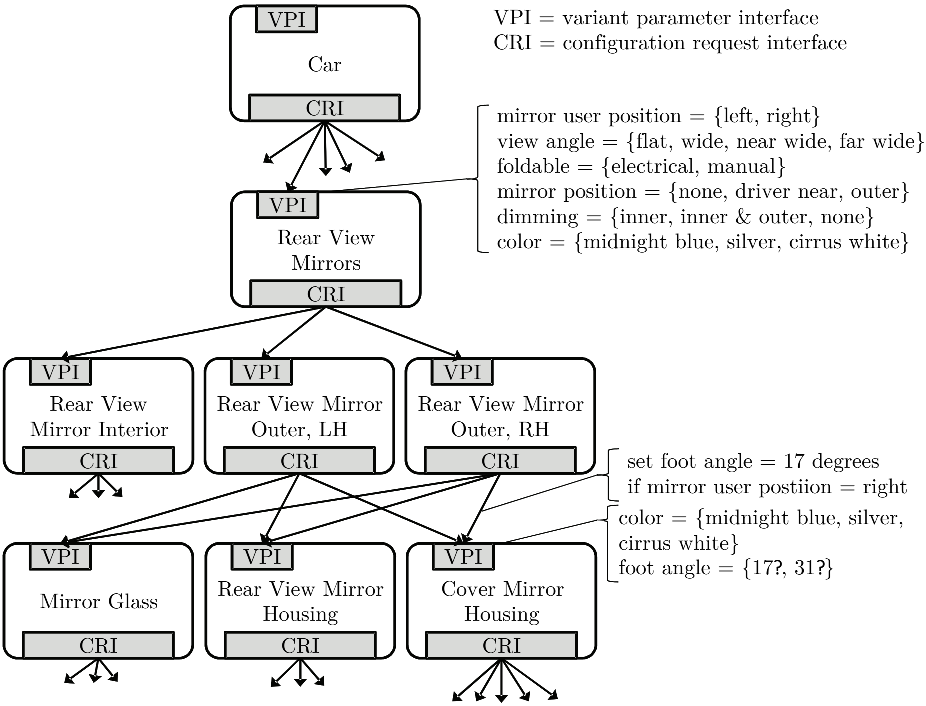


Figure 2.7: The configurable components approach (Claesson et al., 2001).

configuration rules for a configurable component have to reside inside the configurable component.

2.2.5 Configuration rules

All configuration problems may be thought of as constraint satisfaction problems, CSP. The constraints may, however, be structured in an AND/OR graph, which has been shown to be suitable when modelling a set of different but similar product variants (Nilsson, 1980). The structured-based configuration is complementary to constraint-based configuration, and there are also examples of a combination of the two approaches (Günther & Hotz, 1999; Hollmann et al., 2000). AND/OR graphs for a complete set of vehicle configuration rules become too complex to be visualised for the product developers. A subset of the vehicle configuration rules can, however, be represented by AND/OR graphs. As can be seen in Fig. 2.8, both AND (mandatory) and OR (optional) can easily be used for vehicle configuration:

- OR: Alternative items for the *front wiper* to have either an *R/L sensor* OR an *interval wipe*.
- AND: The interior control has to have controls for *wiper* AND *roof*.

The use of logic operators provides a classification of the configuration rules. The configuration rules with IF-THEN statements are called inclusions, while NOT statements are called exclusions (Euwe, 1993).

Logic operators exist in two recognised families of logic: propositional and predicate logic. It is the use of quantifiers that distinguishes propositional logic from predicate logic. Propositional logic uses NOT, AND, OR, IF-THEN and IF-AND-ONLY-IF

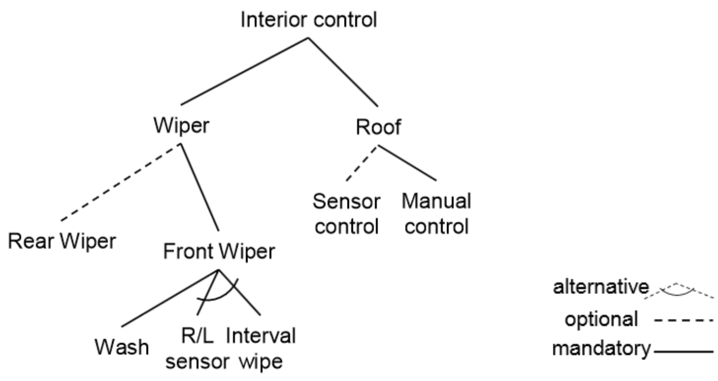


Figure 2.8: The configurable components approach (Bühne et al., 2004).

(Büning & Lettmann, 1999). Predicate logic use the set of logic operators, but also have the quantifiers less than (<), equal to or less than (≤), equal to (=), not equal to (≠), more than (>), equal or more than (≥), the existential quantifier *there is* (∃) and the universal quantifier *for all* (∀) (O'Donnell et al., 2006). There are other logic families that also incorporate the evolution of configuration rules, e.g. the description logics suggested by Krebs (2009). However, no tests with product developers, nor testing with industrial vehicle configuration rules, have been conducted with Krebs algorithms. It is therefore impossible to draw any conclusions regarding the usefulness for the development of vehicle configuration rules.

2.2.6 Conclusions

The configurable product structures that have been reviewed all show evidence of configuration rules. However, the configurable product structures reviewed are mainly conceptual, and not easily formalised into a CSP which would enable computations with algorithms. This research need would be addressed by generalising vehicle product structures, limiting them to the elements that are necessary for a CSP.

2.3 CR development process

The aim of this section is to describe frameworks for the CR development process. The three-step development process is divided into three sections, starting with authoring of configuration rules, continuing with evaluation of configuration rules and then release of configuration rules. Three variations of the CR development process are subsequently discussed. Finally, conclusions are drawn.

2.3.1 Process framework for developing CR

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, 1994). The development of knowledge-based systems includes a process for capturing and transforming knowledge from informal to formal knowledge called the *knowledge acquisition process.* The CR development process is a specific knowledge acquisition process, where the knowledge is represented by configuration rules. The process activities related to the knowledge acquisition are called *authoring* of configuration rules. The following section describes the authoring process in IDEF0 model terms and gives an example of how it takes place.

2.3.2 Authoring of configuration rules

The IDEF0 diagram for the authoring process is shown in Fig. 2.9. As shown in the figure, authoring of configuration rules may be described using the general process for knowledge acquisition presented by Neubert (1993), i.e. the four-step process of *elicitation*, *interpretation*, *formalisation* and *implementation*. These four steps were used in development processes for knowledge-based systems, e.g. the Model-based and Incremental Knowledge Engineering (MIKE) approach (Angele et al., 1998). The four steps describe an evolution from informal to formal description of knowledge, a shift that can also found in the MOKA approach, which stands for Methodology and tools Oriented to Knowledge-based engineering Applications (MOKA, 2000).

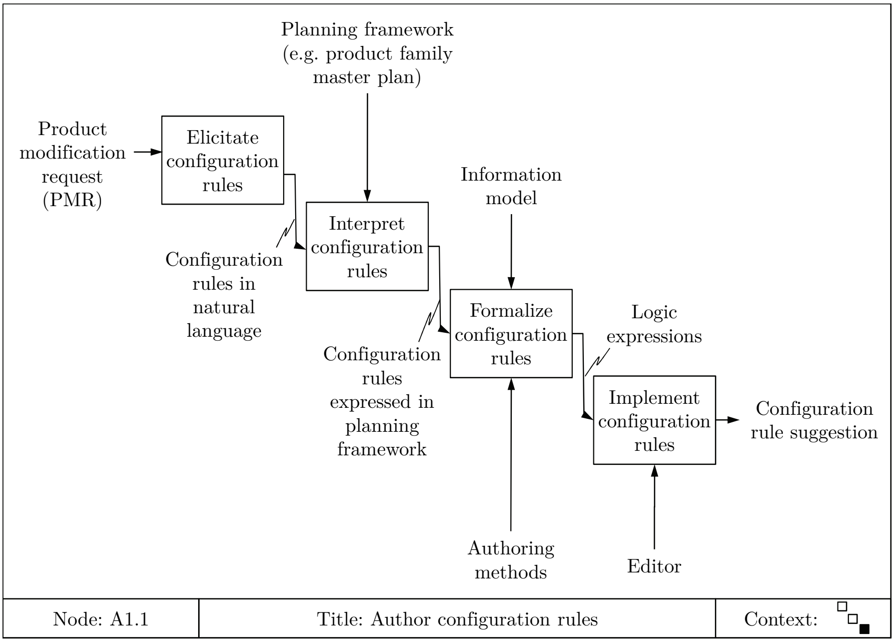
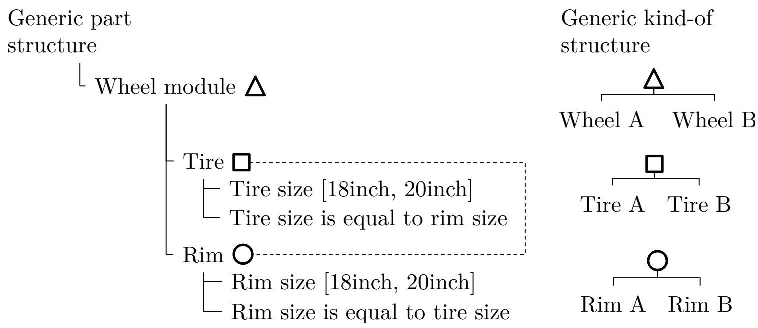


Figure 2.9: Process model for the authoring of configuration rules.

The activities that take place during the authoring of configuration rules are illustrated with an example in Fig. 2.10. Elicitation is the activity of creating an informal description of the configuration rules, often in natural language format. The figure shows an example of a tyre which always has the same diameter as the rim, either 18 or 20 inch. The natural language phrasing of the configuration rules may then be interpreted with the aid of a planning framework, e.g. a product family master plan (Kvist, 2010). Other types of planning frameworks may be matrices and trees that are used for planning the configuration rules. In the figure, the configuration rule is indicated in the tree visualisation by a dotted line. From the planning of configuration rules, it is possible to formalise the configuration rules, i.e. use logic expressions (IF-THEN, NOT, AND etc.). Finally, the formalised rules may be implemented in an information system, e.g. a configurator, by using en editor.

A problem often cited as occurring during the authoring process is the *knowledge acquisition bottleneck* (Schreiber et al., 1993). It has been found to be very difficult to extract and formalise the necessary information from the people in possession of the knowledge. The knowledge acquisition bottleneck is addressed in this thesis through a study of the formalisation step and its use of authoring methods.

1. **Elicitation:** The tyre should have the same diameter as the rim, either 18 or 20inch.
2. **Interpretation:**



3. **Formalisation:** NOT(20tyre AND 18rim), NOT(18tyre AND 20rim)
4. **Implementation:** Typing –(20tyre+18rim) and -(18tyre+20rim) in the editor

Figure 2.10: The authoring process of configuration rules.

2.3.3 Evaluation of configuration rules

The configuration rule suggested by the authoring process is the input for the evaluation process; see Fig. 2.11. The activities for evaluating 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 been applied here 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.

The inspection of configuration rules takes place by visualising the configuration rules, e.g. for design engineers of frame suspensions, brakes, engines etc. The inspection detects mistakes in the configuration rules or modification needed. Either more configuration rules are inspected or the process moves forward to the computations. Computations are performed to further analyse the configuration rules. Complex chain effects of configuration rules are difficult to assess without computations. Iterations to the authoring process take place if any faulty or missing configuration rules are discovered as a result of the computations. The empirical

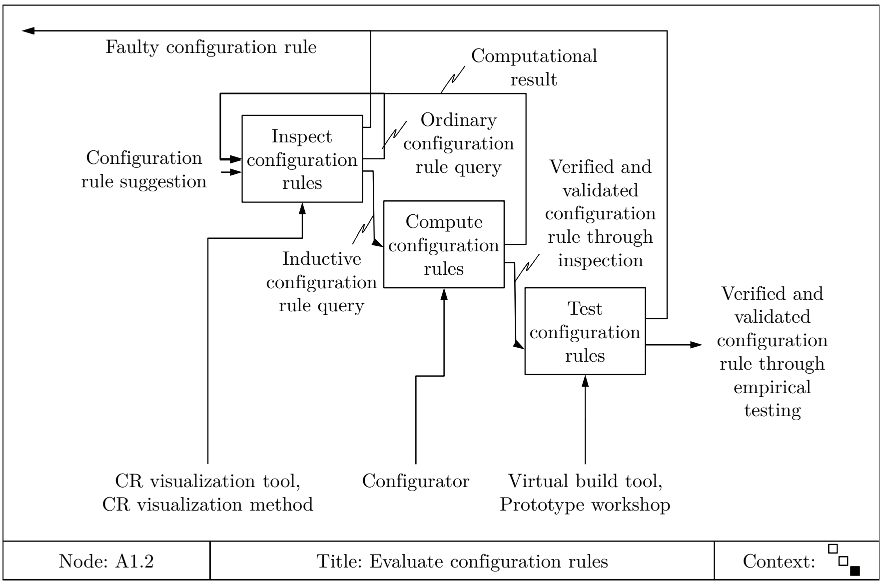


Figure 2.11: Process model for the evaluation of configuration rules.

testing is then used to test the configuration rules by executing them on sample product configurations and then building these product configurations either virtually or physically. Virtual builds for empirical testing of sample vehicle configurations are 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 sets of vehicle configuration rules, so empirical testing only studies a subset of all allowed vehicle configurations. The configuration rules are then ready to be released.

2.3.4 Release of configuration rules

The CR development process as described in this thesis starts with an approved product modification request. The aim of the CR development activities is to implement the change, e.g. a modification of the configuration rules. Once the engineering change has been implemented, it is released for the next phase, the "manufacturing implementation of change" (VDA, 2010). When the configuration rules are released, it is of the greatest importance that there are no errors. The cost of modifying released configuration rules is far beyond the cost of modification of unreleased configuration rules during the development phase. Central concepts that are addressed during the CR development process in order to avoid errors are verification and validation, which are the topic of the next section.

2.3.5 Verification versus validation

In development of knowledge-based systems, the terms verification and validation have definitions that will also be used for this PhD thesis. The distinction between validation and verification used is that verification fulfils some stated requirement, such as every vehicle has one steering wheel – while validation is about developing the right vehicle, e.g. every vehicle has the correct steering wheel (Juristo & Morant, 1998). Verifications have a high potential for automation, while validation does not have a formalised requirement: how does the user know that it is the right steering wheel? The automated verification of knowledge bases has been studied thoroughly in the past (e.g. by Preece et al, 1998; Gupta, 1991; Ayel & Laurent, 1991). Sinz et al. (2003) describe how formal methods may be applied for verification of configuration rules, but the authors admit that in industrial applications the validation activity requires additional knowledge not yet formalised.

Domain specialists (Tuhrim et al., 1988) are people with specialist knowledge about a domain, e.g. brakes, wheels, engines etc. 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 using the CR visualisation. 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 them.

According to Mesihovic (2004), many industrial companies have dedicated roles within product development for reviewing the configuration rule suggestions from the domain specialists. In this PhD thesis, these people are called *product structure specialists*. The product structure specialists verify, but do not validate, the configuration rules.

The process framework for the CR development process has now been established in sufficient detail to discuss three variations of the CR development process.

2.3.6 Three variations of the CR development process

There are three basic variations of the CR development process. The CR development process may take place during:

1. The traditional new development process at a vehicle manufacturing company;
2. The traditional new development process at a vehicle manufacturing company acting as a supplier;
3. The product rationalisation process at a vehicle manufacturing company.

The first variation, the traditional new development process at a vehicle manufacturing company, has no further complications. In the second variation, when an automotive manufacturing company is acting as a supplier, the CR needs to be exchanged. This exchange causes difficulties similar to any product data exchange. Three approaches to product data exchange can be found in the literature: (1) use a neutral file, (2) remodel the product data, or (3) use data instance mapping (Markson, 2007). The neutral file approach involves a translation from the native format to a neutral file format and then a translation again to the receiving system (Pratt, 2001). One well-known product information model standard, first published in 1994, using the neutral file approach is the Standard for Exchange of Product model data (STEP) (ISO, 1994). The use of a standardised information model can be efficient when translators are available, as is commonly the case for part geometries. When there are no translators, as for configuration rules, they need to be developed for the specific exchange and may be difficult to justify financially for a single case. Consequently, there is a lack of industrial validation for neutral file formats for configuration rules (Viel, 2003; Hirel & Hug, 2009). The second exchange approach is to remodel the data. This approach means that one of the companies needs to modify its configuration information model in order to be able to exchange the configuration rules. This second approach is often too expensive for collaborative product development, as a re-modelling affects downstream systems, e.g. manufacturing. The third approach, data instance mapping, consequently needs to be the approach applied in practice for exchange of configuration rules. Data instance mapping uses the data instances from Company A to map data instances at Company B, see Fig. 1. A mapping is 'one-to-one' if every data instance from Company B is mapped by *at most one* data instance of Company A (Zeuthen, 1870). As shown in the figure, the mapping is instead of the type 'onto' if a data instance at Company B is mapped by *more than one* data instance from Company A (MacDuffee, 1940). Data instance mapping is an exchange approach where little research has been conducted.

2.3.7 Conclusions

The CR development process benefits from the KBS development frameworks and terminology. The literature on configuration has been mainly focused on verification of configuration rules, which is an activity that is relatively easy to automate due to its explicit requirement for verification. In order to validate the configuration rules more efficiently, there is a need to go into further details about each method used during the CR development process, i.e. authoring methods, visualisation methods etc. The theory of visualisation methods is described in the following section.

2.4 Configuration systems

System goals are attained by giving feedback on system output. Goals and feedbacks for the system perspectives included in the comparison are shown in Table 2.2. The system managing product configuration rules is called *Product Configuration System* (PCS). As can be seen in the table, either the system goals are for re-using something, or they are to support collaboration. The system feedbacks are found to be not only system output measurements, e.g. from the production system as well as for the PCS system, but also from change management processes, and

Table 2.2: Comparison of goals and feedbacks for adjacent system perspectives on configuration rules development (marked with bold borders).

Configuration systems	**Purpose**	
Technology platforms (Nobelius, 2002; Shapiro, 2006; Meyer & Lehnerd, 1997)	Re-use of...	...technology
Configurable components (Claesson, 2006)		...design knowledge
Product platforms (Kahn, 2012; Wheelwright & Clark, 1992)		...components
Product Data Management (Peltonen, 2000; Guyot et al., 2006).	Support collaboration and time evolution of data.	Supply production with product data.
Product Configuration System (Forza & Salvador, 2002; Leonard-Barton, 1998).		Automate product configuration.
Software Configuration Management (Estublier et al., 2005).		
Knowledge-based engineering (Penoyer et al., 2000; Pinfold & Chapman, 1999)	Automate design tasks.	

physical/virtual/functional testing during development. Each system perspective will now be discussed in terms of goals and feedback.

The economic benefits of *platforms* are rigorously examined in research (Jiao et al., 2007). What platforms have in common is the goal of systematic re-use: a *technology platform* has the aim of re-using technology (Nobelius, 2002), the *configurable components* approach *has the* aim of re-using design knowledge (Claesson, 2006) and a *product platform has the* aim of re-using product components. According to Shapiro (2006), the motivation for using a technology platform is to meet the challenges of a diverse product portfolio where components cannot be re-used. Meyer and Lehnerd (1997) refer to the technology platform as the generic capabilities and knowledge concerning technologies that a company uses to develop its product platforms. According to the PDMA handbook (Kahn, 2012), a product platform is defined as the components that are shared by a set of products. The goal of configurable components (Claesson, 2006) is to re-use not only product components but design knowledge for a product platform. This is achieved by including modelling of design rationales and functional requirements. Levandowski (2012) refers to configurable components as the bridge between technology platforms and part-based platforms. The product families are stored in an information system called Product Data Management (PDM) system.

The use of PDM systems was initially focused on management of documents, e.g. CAD files. The central database of PDM systems might also manage metadata such as "ownership" 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. 2.12 shows a proposed system architecture for the use of configurators in a PDM system. There are two interfaces, the *developer* interface and the *end-user* interface. The system-developer interface is used for visualising the product structure, e.g. configuration rules and metadata for components (creator, versions, etc.). However, many vehicle manufacturing companies have developed PDM systems in-house: 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).

If the term *product data* was to include all data connected with the products, most of the data managed within a manufacturing company would be product data. The term product data is, however, restricted because of its origin as an engineering term (Peltonen, 2000). The term product data is therefore equivalent to engineering data. There are, however, PLM vendors that have product suits including product cost management, e.g. PTC's Windchill Cost (PTC, 2014), Oracle's Agile Product Cost Management (Oracle, 2014) and Dassault's Enovia X-BOM Cost Analytics (Dassault, 2014). In Fig. 2.13, the Primary Cost Estimate is the part cost calculated from the cost model in Windchill Cost. There is a lack of transparency in how this cost is estimated, but PTC's webpage states that the cost model could be based on

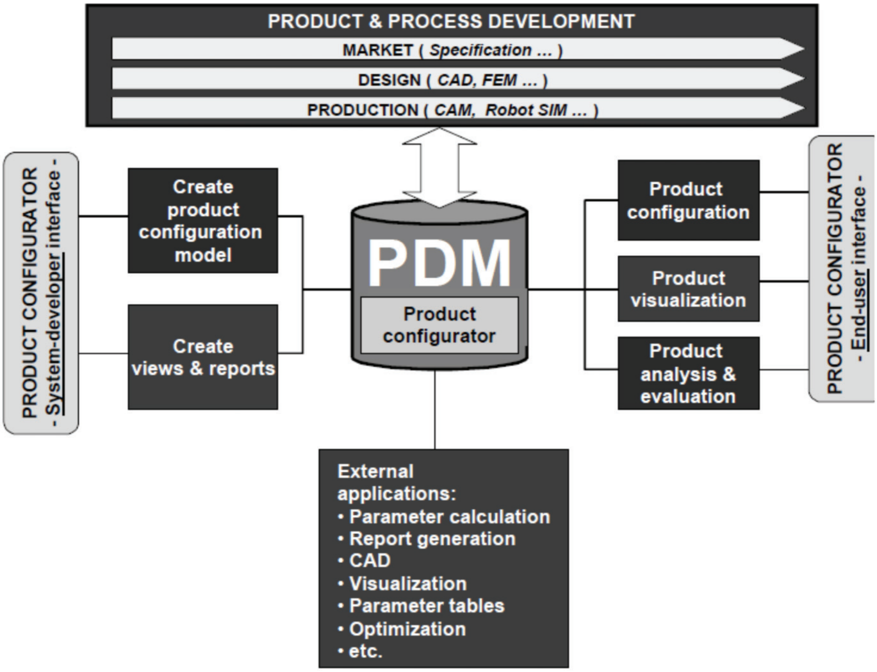


Figure 2.12: Product configurators used in the PDM system (reprinted from Mesihovic & Malmqvist, 2000).

part geometry and thereby production costs. Oracle's Agile Product Cost Management instead puts emphasis on the estimation of part costs during supplier negotiations, i.e. when deciding between outsourcing and manufacturing in-house. The trend is thus that the PLM vendor focuses on sharing part cost information between organisations such as manufacturing, product development and suppliers.

Part Number	Description	Cost Status	Target Cost	Primary Cost Estimate	Expected Volume
LMS700	Display Module 7 inch Diagonal	✓	12	10	100000
SJ2HA40	Glass Touch Screen	✗	11	11.23	25000
PGH67094A	Braces - 4 screw	∅			
PGH4560	Bracket	✗	0.65	0.7	100000
PGH18213	Screw - 5mm	✓	0.15	0.1	400000
SFIN4C2	Flash - eMMC NAND 16GB, MLC	✗	10.5	11.25	25000

Figure 2.13: Part costs as an attribute in PTC's Windchill Cost.

The goal of a PDM system is to be the single source of product data in the production chain. Some of the product data is used to configure the product, and this data is called product configuration data. The system managing product configuration data is called *Product Configuration System* (PCS). A fundamental part of the PCS is a product configurator, which is an application of artificial intelligence techniques for companies selling products adapted to customer needs (Darr et al., 1998; Faltings et al., 1998). The definition of PCS in this thesis is the bundle of product configurator and human and organisational resources that interact with it (Forza & Salvador, 2002; Leonard-Barton, 1998). This socio-technical approach combines the product configurators with human factors such as ease of use. Another approach is to only consider the logical model of the product configurator as a system theory (Waldinger & Stickel, 1992; Sinz et al, 2003). The system approach is then to study a purely technical system. With this more restricted system boundary, it is not possible to study validation of configuration rules, as the validation is currently highly dependent on the users (Hall, 1990). The validation activity is a major activity during the development of configuration rules. The evolution of product data for software is supported with *software configuration management* (SCM). What sets SCM apart from other applications for change management is its focus on files management (Estublier et al., 2005).

Knowledge-based engineering (KBE) is used to automate engineering tasks by applying design rules (Penoyer et al., 2000). Pinfold and Chapman (1999) define KBE as a framework for capturing and defining the process of design creation. The goal of KBE is therefore to increase the degree of automation of engineering tasks in general. This is in contrast with PCS which specifically automates the configuration task.

2.5 CR visualisation

The aim of this section is to show some examples of CR visualisation methods to illustrate the classifications: list-based, table-based, matrix-based and graph-based methods. Finally, conclusions are drawn.

Visualisation supports human beings in dealing with decisions that cannot yet be automated (Wong, 1999). Hence, visualisation is a suitable method when validating configuration rules. A visualisation method is defined as a *systematic and rule-based graphical representation with the aim of acquiring insights, developing an elaborate understanding or communicating experiences* (Lengler & Eppler, 2007). Examples of

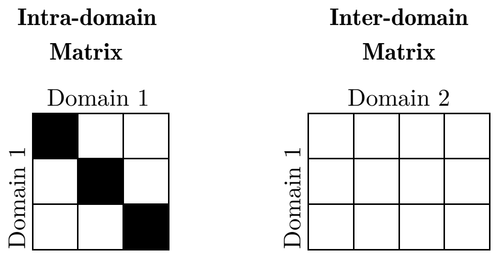


Figure 2.14: Comparison between intra-domain matrix and inter-domain matrix (adapted from Malmqvist, 2002).

visualisation methods are tables, pie charts, mind maps, Gantt charts, decision trees etc. Some of these, which are often used when visualising configuration rules, will be described in the following sections.

2.5.1 List-, table- and matrix-based visualisation methods

Lists, *tables* and *matrices* are used in everyday language, and require some sort of formal definition in order to understand their differences: lists are single-columned tables, matrices have both column and row headings, while tables only have column headings.

Among these three visualisation methods, only the matrix is discussed in configuration research. There is a for example a classification of intra- and inter-domain matrices, where there is a distinction if the column headings are from the same domain as the row headings, see Fig. 2.14. 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 visualise the configuration rules. Vehicle configuration rules within a single domain could be visualised, for example, with the Design Structure Matrix (DSM) (Steward, 1981) or the Quantified DSM (Luh et al., 2011). Vehicle configuration rules requiring both the feature and item domain need an inter-domain matrix, also called a domain mapping matrix (DMM). The K- and V-matrix (Bongulielmi et al., 2002) has three matrices, two DSM and then one DMM, which then together form a multiple-domain matrix (MDM) (Lindemann et al, 2009).

Another visualisation method that is used to visualise configuration rules is the graph, which is the topic of the next section.

2.5.2 Graph-based visualisation methods

A graph can also be used to model relations between elements. A graph consists of *nodes* which are connected through *edges*. The graph can be used to analyse the properties of the relationships, for example the number of edges per node. A software supporting a graph-representation of configuration rules is Loomeo by Teseon, see Fig. 2.15.

Directed graphs, also called *trees*, could represent any configuration rule by going from the tree root to its leaves. This approach has been used for configuration rules, e.g. the Attribute Tree (Schuh & Jonas, 1997). The tree is also a common visualisation method when visualising the modular breakdown structure of a product, i.e. complete product, modules, sub-modules and single items. This approach was used, for example, by the Product Family Master Plan (PFMP) (Harlou, 2006). The PFMP capability of also visualising complete configuration rules in the breakdown structure is, however, limited, and emphasis is instead put on positioning the configuration rules. An undirected graph, also called a *network*, is not capable of describing configuration rules which are more complex than pair relations.

As there are several visualisation methods that could be used when visualising configuration rules, the most important measurement of how good the visualisation

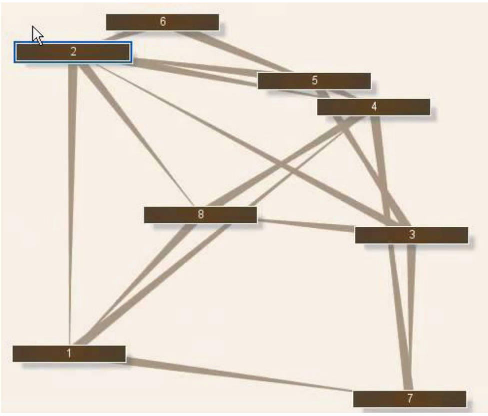


Fig. 2.15. Graph visualised with the Loomeo software (Loomeo, 2011).

method is its user-friendliness. User-friendliness is discussed in the next section.

2.5.3 User-friendliness of CR visualisation tools

Configuration rules are combinations of feature variants, or of feature variants and items. The product model's authorisation rules may authorise the use of a feature variant, a feature variant combination or an item for a certain product model. It is, however, the feature variant combination that is critical from a user perspective, as the number of feature variants in combination may be very high. It has been shown that product developers in the automotive industry can barely even manage a combination of just three feature variants (Hami-Nobari & Blessing, 2005). The configuration rules, however, commonly consist of more than three feature variants, e.g. 40% were found in a study at a car manufacturer (Ohl, 2000). However, to our knowledge there is no literature on the topic of user-friendliness and visualisation of configuration rules. There are however, some characteristics for how the configuration rules are authored that makes their visualisation easier to use. Three examples showing CR visualisation tools can pinpoint these characteristics.

The first example shows a list of configuration rules. As shown in Fig. 2.16, the configuration rules in the figure use a relatively rich set of operators to create the configuration rules (>, =, +, / etc.), which makes it difficult to visualise the configuration rules with anything other than a list.

The second example shows a tree-based visualisation method. As can be seen in Fig. 2.17, no explicit logic is visible, which is similar to matrix-based visualisation methods. The number of leafs in the tree is equal to the number of configurations, which means the tree has a large number of leaves if several feature families are visualised together.

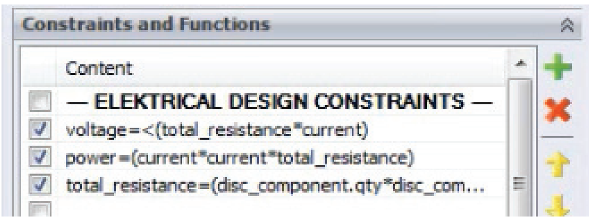


Figure 2.16: List of configuration rules in commercial configurator (Tacton, 2011).

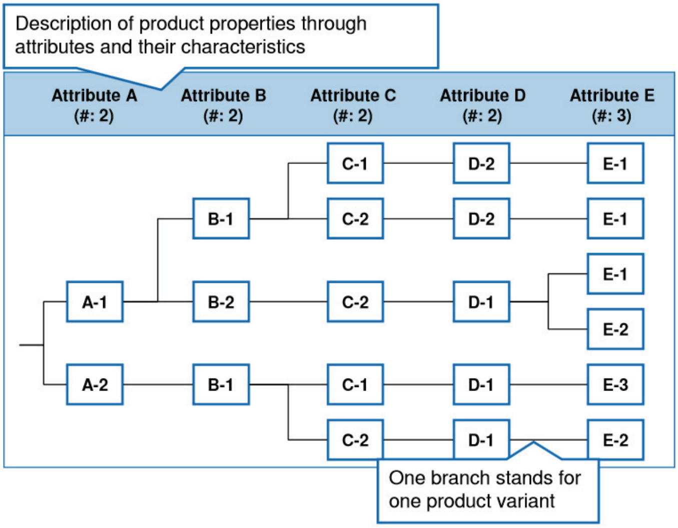
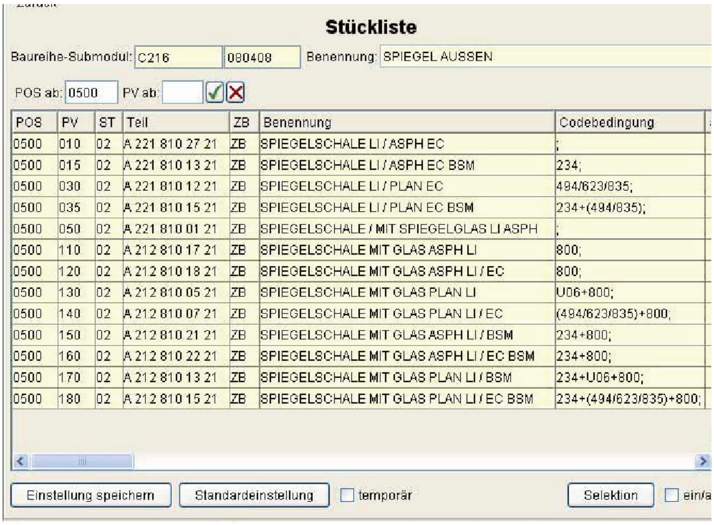


Figure 2.17: The Attribute Tree (Schuh & Kampker, 2010).

The third example shows a table of configuration rules, with increasing user-friendliness because of its columns. The item usage rules in Fig. 2.18 are based on an



POS	PV	ST	Teil	ZB	Benennung	Codebedingung
0500	010	02	A 221 810 27 21	ZB	SPIEGELSCHALE LI / ASPH EC	;
0500	015	02	A 221 810 13 21	ZB	SPIEGELSCHALE LI / ASPH EC BSM	234;
0500	030	02	A 221 810 12 21	ZB	SPIEGELSCHALE LI / PLAN EC	494/623/835;
0500	035	02	A 221 810 15 21	ZB	SPIEGELSCHALE LI / PLAN EC BSM	234+(494/835);
0500	050	02	A 221 810 01 21	ZB	SPIEGELSCHALE / MIT SPIEGELGLAS LI ASPH	;
0500	110	02	A 212 810 17 21	ZB	SPIEGELSCHALE MIT GLAS ASPH LI	800;
0500	120	02	A 212 810 18 21	ZB	SPIEGELSCHALE MIT GLAS ASPH LI / EC	800;
0500	130	02	A 212 810 05 21	ZB	SPIEGELSCHALE MIT GLAS PLAN LI	U06+800;
0500	140	02	A 212 810 07 21	ZB	SPIEGELSCHALE MIT GLAS PLAN LI / EC	(494/623/835)+800;
0500	150	02	A 212 810 21 21	ZB	SPIEGELSCHALE MIT GLAS ASPH LI / BSM	234+800;
0500	160	02	A 212 810 22 21	ZB	SPIEGELSCHALE MIT GLAS ASPH LI / EC BSM	234+800;
0500	170	02	A 212 810 13 21	ZB	SPIEGELSCHALE MIT GLAS PLAN LI / BSM	234+U06+800;
0500	180	02	A 212 810 15 21	ZB	SPIEGELSCHALE MIT GLAS PLAN LI / EC BSM	234+(494/623/835)+800;

Figure 2.18: Table-based CR visualisation tool at Daimler (Lamberti, 2011).

IF-THEN statement, where there is one column for the IF-statement and another column for the THEN-statement. The items (THEN) are listed, but the table is created when adding the item usage rule's feature variant combination (IF).

The last example shows a matrix of configuration rules, which has even higher user-friendliness. As shown in Fig. 2.19, the configuration rules now build a pattern with either filled or empty cells. According to Bongulielmi et al. (2002), the benefit of using a matrix compared to the table and list is that the matrix is able to represent presence or absence of a relationship. Furthermore, there is no explicit logic with brackets or logic operators (AND, IF-THEN etc.), which is another characteristic that increases the matrix's user-friendliness.

The conclusions from the literature review on CR visualisation methods will now be described.

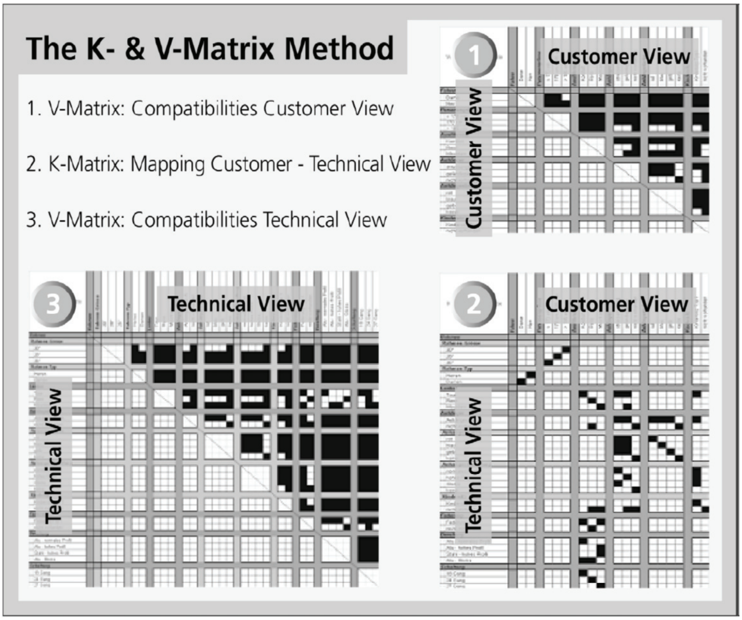


Figure 2.19: The K- and V-matrix (Bongulielmi et al., 2002).

2.5.4 Conclusions

The choice of visualisation methods is to some extent dependent on how the configuration rules are authored. Configuration rule sets may always be defined in lists, but for tables and matrices a certain consistent formalism has to be followed. Tables may, for example, require an IF-THEN formalism in order to add value to the product developers. Matrices have even higher requirements on consistency, as it should be possible to create a pattern with the cells. The major drawback of trees is that they receive a very long list of leaves when the number of configurations increases.

The aim of this section is to provide general descriptions of the most common roles in the product development organisation involved with the development of configuration rules. The aim is also to describe the effects observed from the use of the product structure information systems.

2.6 Organisation

According to (Mesihovic, 2004), many industrial companies have dedicated departments within product development to review requests for new configuration rules from the design engineers. It is the design engineers that are capable of validating the configuration rules. The department for product configuration has no responsibility for ensuring that no configuration rules are missing, but rather analyses and executes the requested modifications to the configuration rules. As can be seen in Fig. 2.20, there are a large number of stakeholders communicating because of the configuration rules, e.g. finance, marketing, manufacturing and product planning. This PhD thesis, however, only considers the interaction between design engineers and product structure specialists.

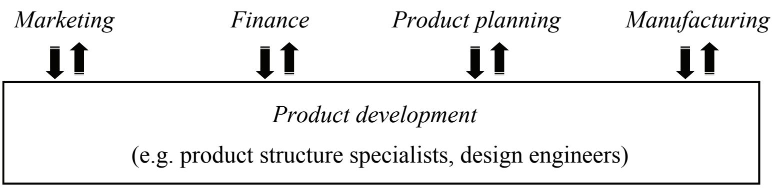


Figure 2.20: Configuration rule specialists and design engineers with their interactions to other functions, adapted from (Mesihovic, 2004).

2.7 Research needs

This PhD thesis contributes to the research areas of CR information modelling, CR development process and CR visualisation. This section discusses the research needs that have been identified. The next chapter will formulate research questions motivated by the research needs.

2.7.1 CR information model

A research need identified by Franke (1998) was the need for a common configuration information model. Three approaches to modelling the configuration information have been identified:

1. **Implementation model.** An implementation model is a reflection of the classes and attributes that are in industrial use for vehicle configuration information. The intention is to be a complete information model, i.e. the level of detail corresponds to the vehicle configuration information that has been created. This complete information model is used, for example, when no information should be lost during an information exchange between information systems. An example of an implementation model is the standard AP214, which has been criticised for being too rich in information, and the benefit of using the standard has not been proven in industrial implementations.
2. **Schematic model.** A schematic model omits all details that are not relevant to the information that it is intended to communicate. The description of the vehicle configuration information model in literature is often schematic in order to illustrate a certain aspect, for example the distinction between the feature and item domains by Haag (1998). Another example is the identification of typical information model classes for vehicle configuration in Mesihovic & Malmqvist (2004). These schemes should illustrate the vehicle configuration information model, but are not as complete as implementation models.
3. **Mathematical model.** A mathematical model is a description using mathematical concepts and language. Mathematical models could be used to describe vehicle configuration, but also to simulate its behaviour. A mathematical model of vehicle configuration is the formulation of a constraint satisfaction problem (i.e. variables, variable values and constraints). Vehicle configuration information from Volvo (Voronov et al, 2011), Daimler (Sinz et al, 2003) and Renault (Astesana et al., 2010a) has been mathematically modelled. The quality of a mathematical model depends on how well the model corresponds to experimental results, i.e. virtual or physical realisations of vehicle configurations.

The three approaches have all been applied in vehicle configuration research. However, vehicle configuration research rarely studies the three approaches in combination. For example, the mathematical models (Voronov et al., 2011, Sinz et al., 2003; Astesana et al., 2010a) have been created without any preceding process analysis.

A schematic model is needed when analysing the CR development process for higher efficiency potential. In order to create a trustworthy support for an efficient CR development process, there is a need for a schematic information model with traceability to both implementation and mathematical models. The schematic model can be created by lowering the level of detail of an implementation model. Furthermore, a more efficient CR development process through the use of computation support requires mathematical modelling. The schematic model should then be reflected in a mathematical model.

2.7.2 CR development process

A recognised problem with the development of variant-rich products is the large amount of testing required. The major driver of complexity, and testing, is the number of feature variants (Weber, 2009). Not all feature variant combinations are offered to the customers, but nevertheless, for example, in 2009 BMW offered the 7-series in $3.5 \cdot 10^{30}$ unique vehicle configurations. Each configuration has to be evaluated both geometrically and functionally. According to Weber (2009), a strategy adopted by Japanese automotive manufacturing companies is to only evaluate the most frequently selected 20% of vehicle configurations. There are two issues when adopting this strategy: 1) it is difficult to predict the 20% most selected vehicle configurations if the vehicle configurations are only sold once or on very few occasions every year, 2) there will be customers that have to wait for a very long time since the vehicle configuration's development might not have been started prior the customer order.

Motivated by the large amount of high effort involved in testing vehicle configurations, a process analysis of the CR development process is needed. In order to make the CR development process more efficient, the process has to be modelled and analysed with development frameworks from, e.g. knowledge-based system (KBS). Time-consuming activities and risk of errors should be better managed with improved CR visualisation tools.

2.7.3 CR visualisation

The most commonly used computer support when developing vehicle configuration rules is the CR visualisation tool. With this visualisation tool, vehicle configuration rules could be visualised with lists, tables, matrices, trees etc. Inspection of the CR visualisation is considered to be time-consuming, but the user-friendliness of the CR visualisation tools has not been thoroughly evaluated in the literature. To fulfil the purpose of this PhD thesis, industrially used CR visualisation tools should be further developed in order to achieve a higher usability and time efficiency.

3 Research Approach

This chapter starts by describing the research questions, which are derived from the research goals and the research needs described in previous chapters. The research setup and process are then described, followed by the research framework. The research process is subsequently described with a discussion of the research methods. Finally, the validation approach describes the actions taken to reduce 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 thesis. The parts of the framework studied concern configurable product structures (*information*), CR development process (*process*) as well as the CR visualisation tools for developing configuration rules (*information system*). Each research question is addressed by more detailed research questions which are to be found in the appended papers.

3.1.1 CR information model

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 research analysed showed that there are attempts to generalise product structures, and standards for how to document vehicle product structures. The attempts to generalise have, however, 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 to lack 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 research needs identified address the issues of the lack of an information model for vehicle product structures, the following research question (abbreviation: "RQ") was stated:

RQ1: What are the characteristics of a vehicle configuration information model 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 both the process modelling as well as the development methods. As the CR development process is iterative, there is a need to describe the complete process in order to understand the difficulties, e.g. why the iterations occur. This thesis approaches this research need by addressing both the process in terms of a literature review on KBS development frameworks, as well as with empirical studies. Three variants of development can be identified: 1) new development processes, and 2) exchange processes between automotive manufacturing companies, and 3) the product rationalisation process. The research question for the literature review was formulated as:

RQ2: How should vehicle configuration rules be developed time-efficiently and error-free?

The CR visualisation tools play a central role in the CR development process, as they are used inspecting and evaluating the configuration rules. The CR visualisation tools are therefore addressed with the research question in the following section.

3.1.3 CR visualisation

The research need for the CR visualisation tools is fundamental, as there is a lack of efficient and user-friendly tools. The CR visualisation methods were approached from scratch by studying which methods are used in the industrial CR visualisation tools. The research question was formulated as:

RQ3: How can CR visualisation tools be improved?

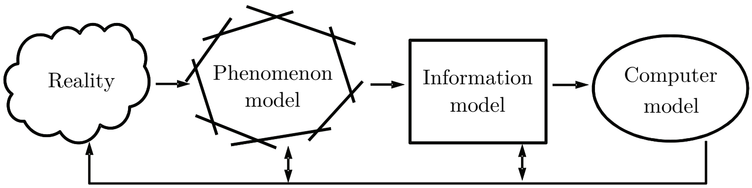
The next sections motivate and describe the methods that have been used during the research process.

3.2 Research setup and process

This research project was conducted with the participation of three vehicle manufacturing companies. The collaboration took place within the framework of FFI, "Fordonstrategisk Forskning och Innovation" ["Strategic vehicle research and innovation", in Swedish], with a vehicle manufacturing company as project leader. This ensured a substantial commitment for the research project within the industry. The project constellation was based on three vehicle manufacturing companies, the two Chalmers' departments "Product and Product Development" and "Signals and Systems", and the Chalmers-Fraunhofer research centre for industrial mathematics. The research project studied vehicle configuration from a holistic viewpoint, e.g. processes, information systems, algorithms etc. The focus of this 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 evaluating efficiency of algorithms.

3.3 Research framework

The research goal can be fulfilled by developing a CR visualisation tool that makes the CR development process more time-efficient and less error-prone. How the CR visualisation tool affects the reality of the CR development process can 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. Where appropriate, these phenomenon models are then developed into information models, which it is possible to utilize in the CR visualisation tools to for support the CR development process. Information models are based on information theory, for example, object-oriented modelling or the unified modelling language UML. Computer models are based on computational



Activity:	Develop vehicle configuration rules.	Model the CR development process.	Generalise the information model.	Implement the formalised methods, Develop visualisation 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 thesis, adapted from (Duffy & Andreasen, 1995).

theories or languages. The important looping arrows within the research framework indicate the iterative refinements that are necessary to develop an efficient tool.

This thesis applied Duffy & Andreasen's (1995) research framework when studying the vehicle CR development process. The "reality" of developing configuration rules was studied by conducting user studies and industrial PDM systems. A phenomenon model defined the CR development process, aided by the theory of knowledge-based systems as well as the process modelling language IDEF0. The generalised 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 visualisation tool supporting the development methods. The CR visualisation tool was developed iteratively 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, there is a belief in design research that any developed tool makes an impact upon the design process itself. If the CR visualisation tool were implemented in the CR development process, the reality of the process would change. There should therefore be effects on the CR development process from an implementation of a modified CR visualization tool. This last step is related to this thesis' general purpose and success criteria: to make the CR development process more efficient and less error-prone.

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

This thesis started with descriptive study I as soon as the project plan was written. The scope of papers sometimes included more than one single type of study (see Fig. 3.2), as well as there were several papers describing a particular study type.

According to Blessing & Chakrabarti (2009), 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:
Q1. What is a successful product?
Q2. How is a successful product created?
Q3. How can the chances of an improved product be increased?

The answer to Q1 is a product that can be manufactured without any difficulties due to faulty configuration rules, with product configurations that have been developed time-efficiently and systematically rationalised in order to increase the product's profitability. The answer to Q2 is a process modelling of the new vehicle development process, a product configuration rule exchange between OEMs and a product rationalisation process. The answer to Q3 is a formalisation of the CR development methods in order to increase automation, and remaining manual tasks should be supported with an easy-to-use visualisation of configuration rules. The measurable success factor is the number of errors, the learnability as well as the time efficiency when using the CR visualisation 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.

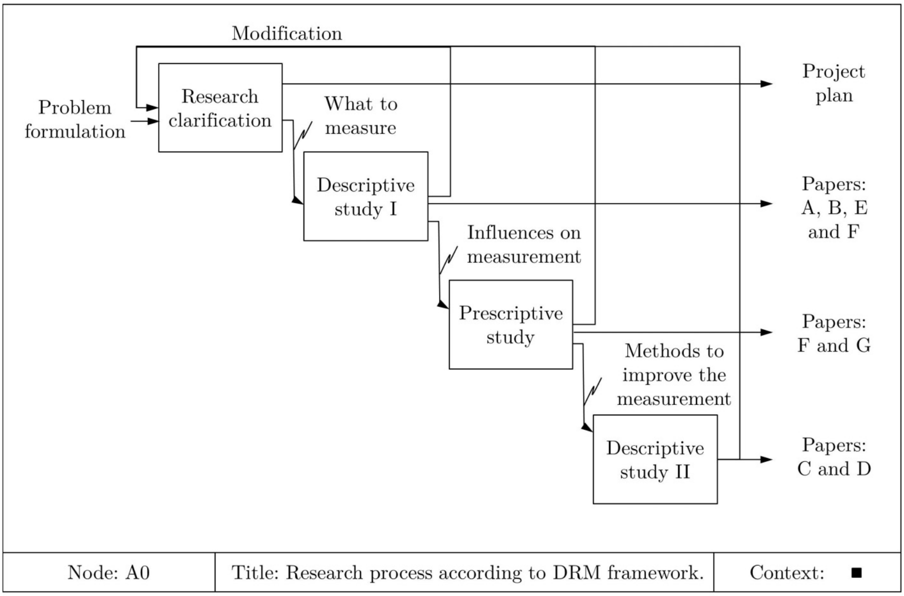


Figure 3.2: Research process following the stages of the DRM framework (adapted from Blessing & Chakrabarti, 2009).

3.4 Validation approach

This section describes the actions to lessen the impact of validity threats for this research project, following the categorisations by Maxwell (2005). How the validity threats have been addressed in the research is described in the following sections, but firstly a short description of each kind of validity threat will be given:

- **Researcher bias:** There are two ways that researchers might be biased (subjective), either by selecting data that fits 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 by, for example, 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-term involvement and intense interviews enable collection of “rich” data, which means that the data is detailed and varied enough to reveal a 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 might 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 as support for qualitative claims as well as in assessing the amount of evidence in the data.
- **Comparison:** Comparing the results obtained with existing results, between control groups or at different points in time may contribute to the interpretability of the results.

The discussion will now concern how the research presented in this thesis’ addresses the validity threats.

The researcher bias is a validity threat that is always critical when conducting research within a department with certain traditions or at companies which have certain approaches to product development. The threat of bias in this research has partly been addressed by discussing different perspectives on configuration, which was done in the Frame of Reference, where, the configuration perspective selected was justified by the purpose of this PhD thesis and not a particular research tradition or single company approach.

Reactivity was systematically questioned by comparing the interview responses to user observations. For example, on one occasion, the configuration rules were judged to be “easy” to develop by an interviewee. That interviewee was then asked to demonstrate what he did on the last occasion that he developed a configuration rule.

From that observation, it was possible to identify certain development tasks where the interviewee needed training.

Other validity threats were also addressed, e.g. intensive long-term involvement, rich data, respondent validation, triangulation, quasi-statistics and comparisons. Intensive long-term involvement was possible as the author of this PhD thesis has been employed at two of the participating automotive manufacturing companies. As it was then possible to have full access to various data sources such as information systems and documents, the use of triangulation has been applied to Papers A-E. As the interviewees were colleagues, it was possible to interview them frequently and hence rich data was collected. Furthermore, as two companies were studied in depth, it was also possible to make detailed comparisons. Respondent validation was achieved by not only interviewing, but also presenting the conclusions from the interviews to the interviewee. Quasi-statistics were important when evaluating the time spent on developing vehicle configuration rules, and how much time could be saved by automating its process activities.

Intervention was also used when creating modified prototypes of the industrial CR visualization tools. It was then possible to test whether the research results actually produced a more user-friendly user interface for the product developer. With the prototypes, it was also possible to search for negative cases where product developers criticised the new user interface.

3.4.1 Research methods

A summary of the research methods used in Papers A-G is given in Table 3.1. As seen in the figure, there is a potential for triangulation since the papers use a variety of research methods. Triangulation strengthens the evidence of the research results. For example, Paper C concerning the development of industrial CR visualisation tools has applied three research methods: “workshop with product structure specialists”, “documentation/information system access” and “formative usability tests (including interviews)”.

Table 3.1: Research methods used for the papers A-G.

	Group interviews	Documents	PDM system study	Interviews	Demonstration/Observation	Usability tests	Literature study	Formalisation
A: Information Modelling for Automotive Configuration	***RQ1***	*RQ1*	***RQ1***	*RQ1*				
B: Authoring and Verification of Vehicle Configuration Rules	*RQ2*	*RQ2*		***RQ2***	*RQ2*			*RQ2*
C: Development of Industrial Visualisation Tools for Validation of Vehicle Configuration Rules	*RQ3*					***RQ3***		*RQ3*
D: Obstacles and Development of Support for Translation of Configuration Rules	*RQ1*	*RQ3*		*RQ3*	***RQ3***			
E: Comparison of Configuration Rule Visualisation Methods		*RQ3*	***RQ3***	*RQ3*	*RQ2*	*RQ3*		
F: A Systematic Process for Developing Vehicle Configuration Rules							***RQ2***	
G: Formal Methods for the Inspection of Vehicle Configuration Rules								***RQ2***

Main research method

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

RQ2: How should vehicle configuration rules be developed time-efficiently and error-free?

RQ3: How can CR visualisation tools be improved?

Paper A: Information Modelling for Automotive Configuration

The results of Paper A are a vehicle configuration information model and a comparison between different product structure theories. The results were achieved by firstly outlining a draft of a vehicle configuration information model. An empirical study was subsequently conducted at an automotive manufacturing company. Since the author has had an intensive long-term (~1 year) involvement at the company studied, there was great potential to collect “rich” data. As shown in Table 3.1, the information sources for the empirical study were:

- *Workshops:* A work group was established consisting of industrial representatives from three automotive manufacturing companies.
- *PDM system study:* It was possible to analyse the PDM system itself when creating a vehicle configuration information model. Attention was given to not considering company-specific information classes.

More details about the research methods can be found in Paper A, which is in the Appendix.

Paper B: Authoring and Verification of Vehicle Configuration Rules

The result of Paper B is a description and formalisation of the authoring and verification methods found at three automotive manufacturing companies. As shown in Table 3.1, the main research method was:

- *Interviews:* A total of 20 semi-structured interviews with product developers, both product structure specialists and design engineers, were conducted. The interviews lasted approximately two hours each. With the aim of reducing the influence of the interviewer, an interview guide was produced and the interviewee was encouraged to speak freely about his/her opinions instead of trying to guess the “right” answer to the interview questions. The interviews were carried out by one or two researchers together with one employee from one of the three automotive manufacturing companies studied. The benefit of being two interviewers was that it was possible to make an analysis directly after the interview, to compare interpretations about what was said and draw some preliminary conclusions.

More details about the research methods can be found in Paper B, which is in the Appendix.

Paper C: Development of Industrial Visualisation Tools for Validation of Vehicle Configuration Rules

The results of Paper C are an evaluation of how the current CR visualisation methods may be extended in order to make the CR development process more time-efficient and less error-prone. As shown in Table 3.1, the main research method for the empirical study was:

- *Usability tests:* A demonstrator was created based on an extended CR visualisation method, implementing the proposed solution during the interviews for Paper B. Four users were selected for the formative usability tests. This follows the formative usability test guidelines, to not conduct large numbers of experiments but to extract as much information as possible from every user. The test cases were based on industrial vehicle configuration rules and real users for the application, e.g. design engineers and configuration rule specialists.

More details about the research methods can be found in Paper C, which is in the Appendix.

D: Obstacles and Development of Support for Translation of Configuration Rules

In order to identify obstacles during a vehicle configuration rule exchange, this paper describes the exchange process in detail. Two automotive manufacturing companies, which were exchanging vehicle configuration rules in an on-going development of a new car, were studied. As shown in Table 3.1, the main research method for the empirical study was:

- *Demonstration/Observation:* As the author was employed at one of the automotive manufacturing companies studied, it was possible to observe the exchange process. The product developers were also able to demonstrate what they had done in previous vehicle configuration rule exchanges. This made it possible to identify what the main obstacles were.

More details about the research methods can be found in Paper D, which is in the Appendix.

E: Comparison of Configuration Rule Visualisation Methods

The aim of this paper is to evaluate the effects of using a certain configuration rule method, especially the impact on how the configuration rules are authored. This was achieved by comparing configuration rule visualisation methods from two automotive manufacturing companies. As shown in Table 3.1, the main research method for the study was:

- *PDM system study:* The configuration rule visualisation methods studied were a matrix and a list. These two configuration rule visualisation methods could be compared as the two automotive manufacturing companies studied were exchanging configuration rules, and therefore used both the visualisation methods.

More details about the research methods can be found in Paper E, which is in the Appendix.

F: A systematic process for developing vehicle configuration rules

This aim of Paper F is to propose a systematic development process for vehicle configuration rules. As shown in Table 3.1, the main research method for the study was:

- *Literature study:* The literature on development methods for product configuration systems and knowledge-based systems was reviewed and compared. The outcome of the literature review then produced the development frameworks and methods needed to create a systematic process model for vehicle configuration rules development.

More details about the research methods can be found in Paper F, which is in the Appendix.

G: Formal methods for the inspection of vehicle configuration rules

Three development tasks conducted with inspection of vehicle configuration rules were addressed with formal methods in this paper. As shown in Table 3.1, the main research method for this study was:

- *Formal methods development:* The three formal methods developed were to 1) reformulate vehicle configuration rules, 2) test feature variant combinations, and 3) count quantities from item sets. These formal methods were put into the context of inspecting vehicle configuration rules, by describing details about visualisation of vehicle configuration rules and the CR development process.

More details about the research methods can be found in Paper G, which is in the Appendix.

4 Results

This chapter presents the main results of the research with an extended summary of the papers.

The papers may be organised into the framework for engineering information management systems:

- Section 4.1. Information modelling CR information model Paper A
- Section 4.2 Process modelling CR development process Papers B, D and F
- Section 4.3 Information system CR visualisation methods Papers C, E and G

The papers will be discussed following their classification into this framework starting with CR information modelling, then process modelling and finally the studies of the information system.

4.1 CR information modelling

One of the most important contributions this PhD thesis makes, is the information model described in Paper A. The information model is an elaboration of the Constraint Satisfaction Problem (CSP) in the vehicle configuration context. The CSP is based on a set of variables that can have predefined values. The information model in Paper A gives names to some of the variables in the CSP, i.e. the *product models*. This information model is then also used as a reference in Paper D, where two automotive manufacturing companies exchange configuration rules and thereby compare information models. The conclusion from Paper D is that there were no issues due to heterogeneous information models, but there were issues as a result of different authoring methods for configuration rules that created both process and information system issues. The recognition of a CSP as an information model among the automotive manufacturing companies opens up the possibility of further studying the creation of translators between company-specific authoring methods.

4.1.1 Paper A: Information Modelling for Automotive Configuration

Significant similarities were found in the product configuration approaches between automotive manufacturing companies. Our hypothesis in this paper was therefore that there is a potential to create a vehicle configuration information model. The 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, specifically considering 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 an automotive manufacturing company, as well as a review of the literature concerning information models.

Results

A vehicle configuration information model was developed; see Fig. 4.1. At the level of abstraction shown in the figure, the model is generally applicable for all vehicle manufacturing companies studied. The top node in the Unified Modelling Language (UML) class diagram is the "product family", including, for example, all vehicles sharing the same platform. The product family might include 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 authorisation 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") combinations of feature variants.

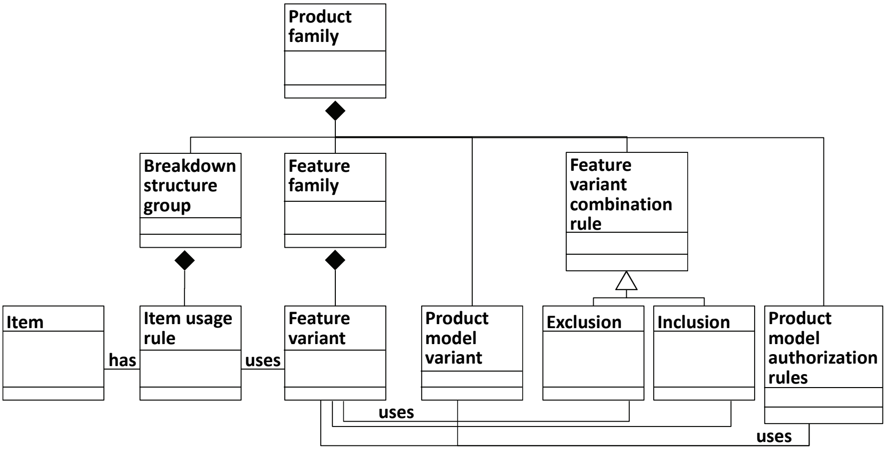


Figure 4.1: UML class diagram of vehicle product structures in Paper A. For UML class diagrams formalism see OMG (2010).

- Item usage rules define for which feature variant combinations a certain item should be used.

The vehicle configuration information model described facilitates the discussions with automotive manufacturing companies. The vehicle configuration information model described is not, however, capable of independently clarify why the development of vehicle configuration rules is time-consuming and error-prone.

Conclusions

Practical challenges during CR development emerge from method- and process aspects rather than information content. Frameworks for product configuration need to go beyond pure information modelling to include methods for writing, e.g. "good" configuration rules in order to increase their utility for industry.

4.2 CR development process

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. Paper B describes the initial results concerning variations in how to author configuration rules. These logically equivalent authoring methods have been known for a long time in computer science. The ambiguity concerning which authoring method to choose is an issue during the CR development process, as different users have different preferences/needs. The variations in authoring methods are also discussed in Paper D as the main obstacle when exchanging configuration rules between automotive manufacturing companies. A more systematic approach for describing the CR development process is described in Paper E, which was based on existing KBS development frameworks.

4.2.1 Paper B: Authoring and Verification of Vehicle Configuration Rules

Paper B had the aim of investigating industrially applied methods for authoring and verification of vehicle configuration rules, specifically to address the difficulties that might potentially lead to faulty vehicle configurations and inefficiencies in the CR development process. The 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. A total of 20 semi-structured interviews were conducted with both design engineers and product structure specialists. The interview sessions included interviewees' demonstrations of authoring and verification methods.

Results
Configuration rules are authored using certain methods, and then verified before release. Both design engineers and product structure specialists are involved in authoring and verification of configuration rules. These roles involve different daily activities, which generate different preferences in authoring methods. The authoring variations that were found are:

- *Overlapping documentation:* 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 can create small “building blocks”, which might then be used when allowed according to the feature variant combination rules.

Conclusions

The literature review showed that the authoring and verification methods for configuration rules described in this paper are rarely studied. This paper fills this research gap by identifying three authoring methods. These authoring methods exist, while there are two kinds of product developers developing the configuration rules: the configuration rule specialists and the design engineers. For example, the use of high-level feature variants is efficient for frequent users, but might be confusing for less frequent users. The less frequent users might not even be aware of the high-level feature variants. Formalisation of authoring and verification methods would provide 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 justifies realising the automation potential and thereby reducing development costs. Furthermore, arguments for using a user interface consisting of a matrix rather than a text-based format were identified. We have shown that the main difficulty is to combine feature variant combination rules with item usage rules, and the traditional user interfaces for CR visualisation tools should therefore be challenged.

4.2.2 Paper D: Obstacles and Development of Support for Translation of Configuration Rules

Inter-organisational exchange of product data commonly entails product data exchange issues (Domazet et al., 2000). Exchange issues due to information models are described in (Fang et al., 1991; Naiman & Ouksel, 1995) as three aspects of information model comparability: (1) “naming”, refers to naming issues for class, attribute or instance, e.g. synonyms, (2) “abstraction” refers to relationships between classes, e.g. “x” is a generalisation of “y”, and (3) “heterogeneity level” concerns conflicts in terms of naming/abstraction of classes, attributes or instance levels.

Another exchange issue that is mentioned in (Fang et al., 1991) is differences in IT systems, for example, how configuration rules are visualised in the two in-house IT systems in the present study. In order to identify obstacles during a configuration rule exchange, this paper will describe the exchange process in detail. The following research questions have been addressed:

RQ1: What does an exchange process for product configuration rules between collaborating companies look like?

RQ2: Which exchange obstacles for configuration rules between collaborating companies are identified?

RQ3: How can the exchange of configuration rules be improved?

An analysis phase included a set of parallel activities: creation of configuration information models, creation of exchange process model, as well as an analysis of automation potential. Then, an algorithm was iteratively developed with a validation during the testing phase.

Results

The exchange process model between the two collaborating companies Alpha and Beta is shown in Fig. 4.2. Beta created a new integration system for the collaboration project, which is used to visualise Alpha's configuration rule matrices, and from which a configuration rule specialist manually detects changes (A3). New feature families and variants are also mapped and new configuration rules authored (A5).

From the process activities with an automation potential (A3-A5), change detection (A3) was addressed with a change detection algorithm. The tests conducted showed that this automation is essential to ensuring the quality of the configuration rules. In addition, the algorithm saves time.

Conclusions

It has been shown that the configuration rule exchange is a process that needs to detect changes, map data instances as well as re-formulate configuration rules. Several types of information model heterogeneity could be identified, however, the heterogeneity in information models was not identified as a major issue. It was instead the heterogeneity in configuration rule visualisation method (matrix vs. lists) that caused the most challenging issues. The conclusion is that tools that are working well for documenting configuration rules within a company do not

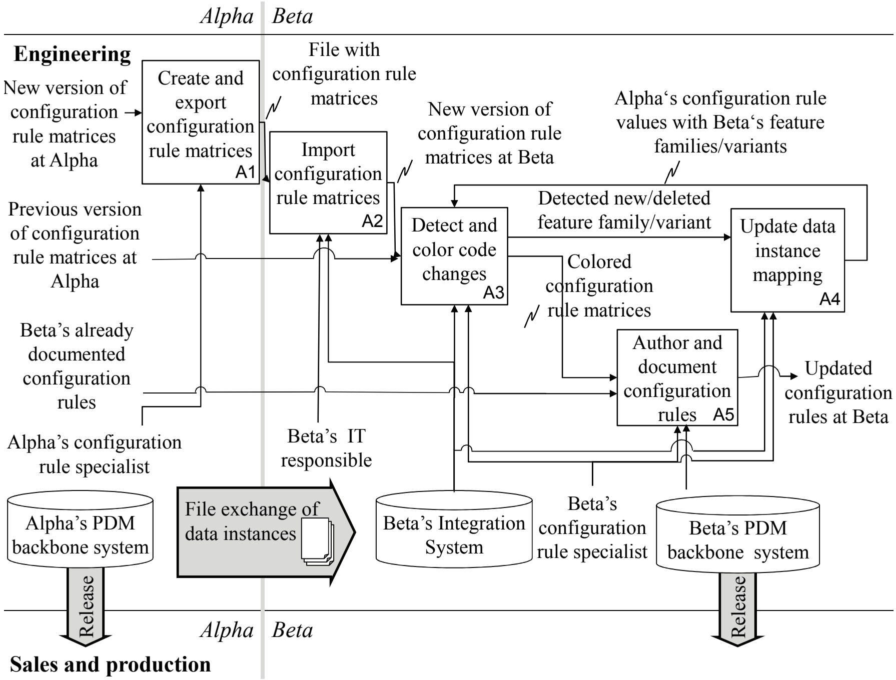


Figure 4.2: Exchange process for configuration rules from Alpha to Beta.

necessarily suffice between collaborating companies. The change detection algorithm implemented for this study serves as an example of what is required to overcome the issue of visualisation method heterogeneity. Future work should address issues identified during the reformulation of configuration rules in order to further improve the exchange of configuration rules.

4.2.3 Paper F: A Systematic Process for Developing Vehicle Configuration Rules

The aim of this paper is to propose a systematic development process for vehicle configuration rules. The approach was to start by conducting a review of the literature on development methods and processes for product configuration and knowledge-based systems (KBS). Empirical studies from the automotive industry subsequently resulted in a model of the vehicle configuration rules development process. This paper's research approach addresses the research question:

RQ: Based on KBS development frameworks, how could the development process for vehicle configuration rules look like?

Results

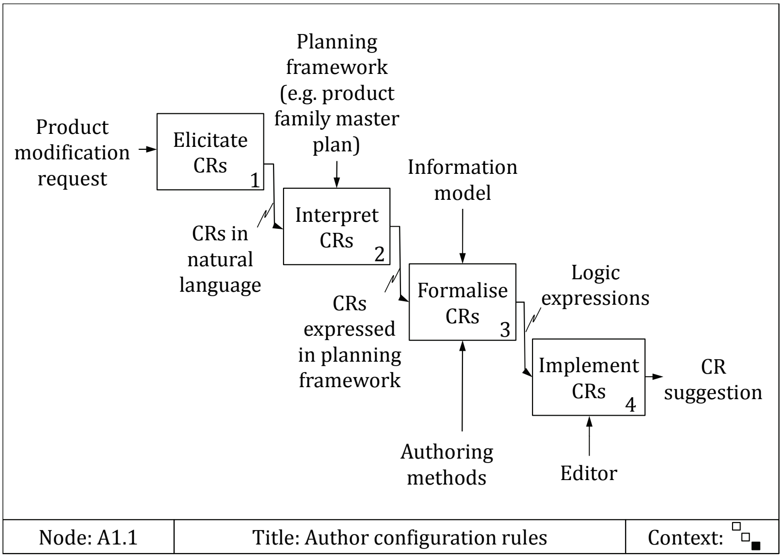
Development of a KBS can be modelled with a two-step framework: the *creation* of knowledge and the *evaluation* of knowledge. The knowledge base for a KBS is created during a process called *knowledge acquisition*. Neubert (1993) models the knowledge acquisition step with a four-step process: elicitate, interpret, formalise, and implement. According to Meseguer and Preece (1995), the second step in the KBS development framework, the evaluation of knowledge, can be classified into three process activities: inspect, compute and test. The application to the development of vehicle configuration rules is shown in Fig. 4.3.

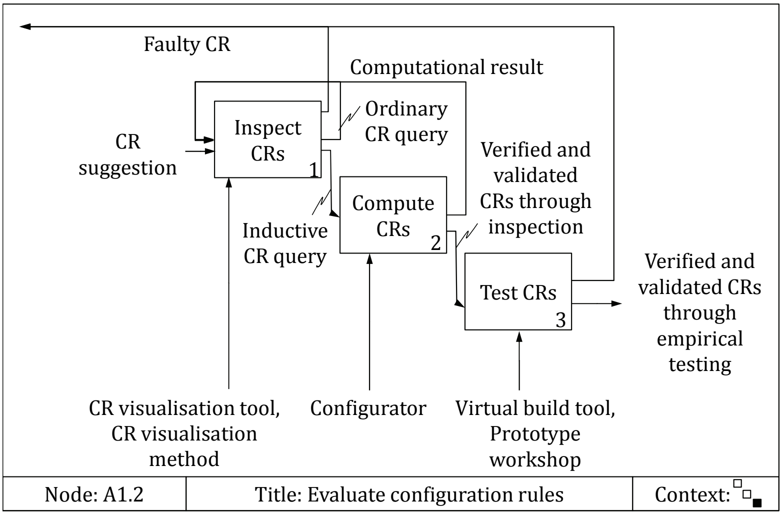
When reviewing the literature, development methods were positioned towards vehicle configuration rule development process activities. The literature review provides several examples of KBS development methods that are identified as applicable and that might potentially improve vehicle configuration rule development. There were also gaps where little research had been conducted (elicitation, interpretation and release). These gaps identified future research needs. Another research opportunity is to study the practical benefit of applying KBS development methods to the development of vehicle configuration rules.

Conclusions

The main result of this paper is the description of a systematic process for vehicle configuration rules development, from its first formulation until production release. The vehicle configuration rule development process includes specific characteristics derived from the presence of product platforms, and the vehicle configuration information model, which then affects the execution of the generic knowledge acquisition and evaluation process. For example, inspection of vehicle configuration rules entails difficulties caused by the three vehicle configuration rule classes, while the difficulties of inspection have been vaguely covered by earlier research in the knowledge-based and “general” product configuration literature.

Figure 4.3: Process models for the authoring and evaluation of vehicle configuration rules.





4.3 CR visualisation

Paper B concluded that the interviewees found matrix-based visualisation methods easier to use compared to list-based visualisation methods. A matrix-based visualisation method which addresses the needs that were found in the interview study is described in Paper C. Paper E then compares visualisations of a configuration rule set by using matrices and lists as visualisation methods. Finally, Paper G describes formal methods for how to increase the support for the product developers by improving the PDM user interface.

4.3.1 Paper C: Development of Industrial Visualisation Tools for Validation of Vehicle Configuration Rules

The conclusions from Paper B stated that the industrial CR visualisation tools needed to combine feature variant combination rules with item usage rules. Based on this finding, Paper C had the aim of finding a CR visualisation tool that is easier to use, thereby making 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 visualisation tools used when validating vehicle configuration rules?

RQ2: Which CR visualisation tool addresses those weaknesses?

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

The pre-study for the development of a CR visualisation tool included a description of current CR visualisation tools, typical configuration rule queries, CR visualisation needs and a discussion of design considerations. A demonstrator was then created to develop and evaluate the CR visualisation method proposed. Formative usability tests were based on industrial data and the participants were real users.

Results

The demonstrator displays item usage rules with black crosses as shown in Fig. 4.4. The item usage rule for Item 1 should be read as: IF(a1 AND b1 AND c3 etc.) THEN(ITEM1). For this item usage rule, there are computations for restricted feature variants that generate the pink fills. Furthermore, fulfilment of exactly one condition for the set of items produces feedback highlighted as question marks ("?").

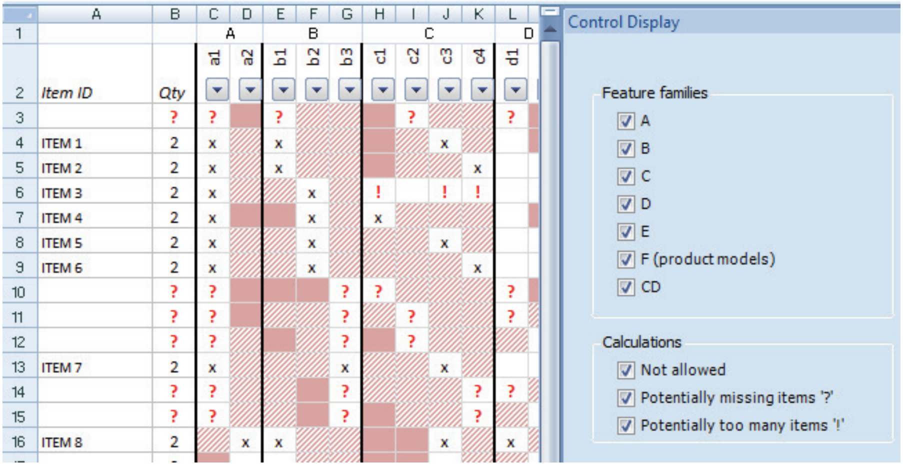


Figure 4.4: User interface of the demonstrator in Paper C.

Also, the exclamation marks ("!") show for which item usage rules there are more than one item from the set of items analysed.

All classes of configuration rules (product model authorisation rules, feature variant combination rules and item usage rules) are thereby visualised in one single user interface of the CR visualisation tool.

The participants in the usability test predicted that the greatest value of the CR visualisation method suggested was the increased confidence of the users. Computations had been automated, e.g. for detecting potentially missing items. This reduced the risk of the making computational mistakes.

Conclusions

The analysis of industrial CR visualisation tools and the related user study have shown that there is a potential to facilitate the CR development process. Usability tests of the demonstrator have shown that it is possible to address the weaknesses identified. 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.
- **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 seconds. Although there was no comparative response time measurement, according to the post-session interviews, time efficiency increased.

4.3.2 Paper E: Comparison of Configuration Rule Visualisation Methods

The aim of information visualisation is to create an effective representation of the information model and the information contained therein (Mackinlay, 1986). Effectiveness means that the user obtains an overview of the information as rapidly as possible. This effectiveness is challenged for the configuration rule visualisation as there might be hundreds of thousands of configuration rules. The main aim of this paper is to compare configuration rule visualisation methods in order to evaluate the effects of using a certain visualisation method, in particular the impact on how the configuration rules are authored. A second aim is to find advantages and disadvantages for the visualisation methods at the two automotive manufacturing companies. The following research questions have been addressed:

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

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

Results

One of the real case examples visualised with Alpha's matrix-based method is shown in Fig. 4.5. Two measurements have been established: a size measurement G in the growth direction (number of columns for the matrix, number of rows for the list), and a combinatory difficulty measurement C which is the number of feature variants in a configuration rule. The same example with Beta's original authoring of configuration rule list is shown in Table 4.2. It now becomes evident that the list has a higher combinatory difficulty, but grows at a lower rate.

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

Number of configuration rules:

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

Fig. 4.5: Alpha's configuration rule matrix.

Table 4.2: Beta's configuration rule list.

IF: Feature variant	THEN: Feature variant combinations	Number of configuration rules	Feature variants in a configuration rule
a1	(*n1* AND *s2*) OR (*n1* AND *s1* AND *t1*) OR (*n2* AND *o5*) OR *n4* AND *p2*	1	10 *(a1, n1, s2, n1, s1, t1, n2, o5, n4, p2)*
a1	*n4* AND *r1*	2	3 *(a1, n4, r1)*
a1	(*n1* AND *s1* AND *t1*) OR (*n4* AND *o4*)	3	6 *(a1, n1, s1, t1, n4, o4)*
a1	((*n1* AND *s2*) OR (*n1* AND *s1* AND *t1*) OR (*n2* AND (*o5* OR *o6*) OR *n4*) AND *q1*) AND *p1*	4	12 *(n1, s2, n1, s1, t1, n2, o5, o6, n4, q1, p1)*
a1	*n3* AND (*o1* OR (*o3* AND *t1*))	5	5 *(n3, o1, o3, t1)*

Conclusions

The visualisation method strongly influences how the configuration rules are authored, as there are different optimisation goals for readability (small number of matrix columns vs. small number of list rows) and availability of logical operators (combinatory difficulty increases with OR operator in the lists). However, the configuration rules specialists interviewed strongly preferred the matrix-based visualisation method even when acknowledging the strength of the list-based visualisation method. The main argument was that the matrix-based visualisation kept the Boolean algebra simple as it did not allow any OR operator. With a simple Boolean algebra, the configuration rules become easier to read for all users working with configuration rules, which would make the development of configuration rules more efficient.

4.3.3 Paper G: Formal methods for the inspection of vehicle configuration rules

In this paper we address three challenges when inspecting vehicle configuration rules: 1) reformulation of vehicle configuration rules, 2) testing of feature variant combinations, and 3) counting of item quantities from an item set. These three challenges have been addressed using algorithms in (Sinz et al., 2003) and (Astesana et al., 2010a,b), however, in this paper they are more comprehensively and systematically approached. The research question addressed is:

RQ: Which algorithms can make the inspection of vehicle configuration rules more efficient when it comes to
a) reformulation of vehicle configuration rules,
b) testing of feature variant combinations, and
c) counting of item quantities from an item set?

The research approach started with a review of the literature, followed by algorithm development. The development of algorithms consisted of three steps: first a description aided by Venn diagrams, then a description with formal methods, and finally an implementation into a matrix-based visualisation of vehicle configuration rules. The last step in the research approach was to test the computational feasibility of the proposed algorithms on three industrial vehicle configuration rules sets.

Results

The algorithms proposed in this paper *test feature variant combinations, count item quantities from an item set* and *reformulate vehicle configuration rules*:

- *Test feature variant combinations:* A feature variant combination has fewer feature variants than a complete vehicle configuration. Product developers normally have some feature variant combinations they use to inspect whether configuration rules are giving expected results. This inspection activity can be automated to some extent if these feature variant combinations are stored as reference configurations.
- *Count item quantities from an item set* is done by analysing whether a set of items fulfils *at least one, at most one or exactly one* conditions. The exactly one condition must be satisfied for steering wheels, chassis, cabin, windscreens, etc. The Venn diagram for this condition is illustrated in Fig. 4.6 and Fig. 4.7. Visualisation of the exactly one condition is also illustrated with a matrix-based visualisation in Table 4.3, where one allowed feature variant combination does not have any item from the item set (ITM001 and ITM002).
- *Reformulation of configuration rules:* A product developer might want to visualise a configuration rule with another set of feature variants. The reformulation only affects how the configuration rules are visualised to the user.

Fig. 4.6: Some allowed configurations with one item, some with two, and some with none from an item set.

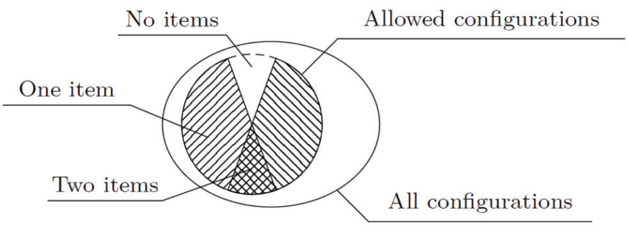


Fig. 4.7: All allowed configurations have exactly one item from the item set.

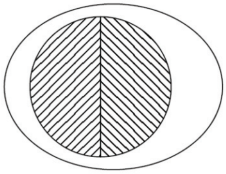


Table 4.3: Visualisation of item usage rules for an item set that does not fulfil the exactly one condition. An allowed feature variant combination without an item is marked with “no item!”.

	Feature variants				
Item ID	1.6L	Turbo	Without Turbo	Gasoline	Diesel
ITM001	x	x		x	
ITM002	x				x
No item!	x		x	x	

Conclusions

The purpose of this paper is to show some examples of how the inspection of vehicle configuration rules may be supported with formal methods. Three formal methods that could be implemented in a visualisation of vehicle configuration rules have been proposed: 1) test feature variant combinations, 2) count quantities from item sets, and 3) reformulate vehicle configuration rules. It has been shown that it is possible to visualise the result of these formal methods during the inspection of vehicle configuration rules. The evaluation of the suggested formal methods gave that they are well suited for implementation, both because of computational efficiency and feasibility for implementation in a visualisation of vehicle configuration rules. Future work involves creating a prototype with the formal methods implemented in order to further develop the visualisation together with product developers.

5 Discussion

This chapter discusses the research results in terms of fulfilment of research goals, answers to research questions and research contributions. The aspects of generalisation, transferability and validity are then discussed.

The general purpose of this PhD 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. This purpose was broken down into three research goals, the fulfilment of which is discussed in the next section.

5.1 Answers to research questions

Each of the research questions is answered with a summary of the research results.

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

The research work for this PhD thesis describes the variance of a product family of vehicles by means of a *vehicle configuration information model*. This information model contains the classes for creating a vehicle configuration, i.e. an instantiation from the product family. The first step in a vehicle configuration is to select the *product model,* e.g. *Volvo XC90* or *S60*. The product model authorises a list of feature variants to be selected as a second step during the vehicle configuration. Some selections of feature variants are incompatible, i.e. their combination is not acceptable. In the third step of the vehicle configuration, the selection of feature variants is checked in order to make sure that the requested feature variant combination is buildable. The list of items required to manufacture the vehicle is generated from this, possibly modified, list of feature variants. Each of the three steps of the vehicle configuration uses a class of configuration rules. When the customer has selected a product model, there are *product model authorisation rules* that define a list of permitted feature variants. The compatibility of feature variants is checked with *feature variant combination rules*. Finally, a list of items is created from the combination of feature variants by using *item usage rules*. For more details about the vehicle configuration information model see Paper A, p. 4, Fig. 1.

As stated by Haag (1998), a characteristic of the vehicle configuration information model is the existence of two domains: *feature* and *item* domain. Haag does not discuss the *configuration rules* which exist both between (inter) and within (intra) these domains, see Paper A, p.4, Fig. 1. The inter-domain configuration rules are called item usage rules, and the intra-domain configuration rules are called product model authorisations and feature variant combination rules. A literature review in Paper A showed that the item usage rules in particular were missing from half the literature studied: Feature models (Bühne et al., 2004), Probabilistic feature models (Czarneck et al., 2008), Product configuration view to software product families (Männistö et al., 2001), Generic BOM (McKay et al., 1996), QFD method (Akao, 1994), Complexity Manager (Schuh Group, 2014) and Software Configuration Management (Di Cosmo, 2006). Literature also existed describing the existence of item usage rules: Configurable components (Claesson, 2006), System-based product modelling (Collier, 1999) and K- & V-matrix method (Bongulielmi et al., 2001). None of them do it formally, i.e. by using logic expressions.

The reason that vehicle configuration information models have to be based on both the feature and the item domain originates from the number of items. The customer may find it impossible to configure a vehicle based solely on the list of items (tens of thousands of items). An item corresponds to a part, a sub-assembly, an interface between parts, an assembly instruction document etc. Other products families with numerous items may have these two domains as well.

Another characteristic of the vehicle configuration information model is its logic-based configuration rules. The configuration rules are authored by using logic operators such as AND, NOT, OR, IF-THEN etc. This is in contrast to *parametric design rules*, which are necessary for design-to-order or engineer-to-order products. Vehicle configuration is fully based on pre-designed items, which thus does not justify use of parametric design rules. Parametric design rules are similar to mathematical equations that use a parametric design that is modifiable. This characteristic of having logic-based configuration rules affects the CR visualisation method. Logic-based configuration rules are easily processed in order to be visualised according to a specific CR visualisation method. It is, however, not possible for parametric design rules to be processed and hence have a more limited scope for their applicable CR visualisation method. Only list-based design rules were found in the research work for this PhD thesis, e.g. Tacton (2013). The vehicle configuration information model clearly states that the vehicle configuration rules are based on inclusions (IF-THEN) and exclusions (NOT), which consequently excludes design rules.

The vehicle configuration information model thus needs to be capable of defining two important characteristics: 1) three classes of vehicle configuration rules required for configuration of the feature and item domains, and 2) configuration rules authored with logic operators (NOT, AND, IF-THEN, etc.). The vehicle configuration information model described is a special case of a constraint satisfaction problem, CSP. The algorithms for solving the CSP, or configuring the vehicle, are consequently available from previous research, e.g. Sinz et al (2003) and Astesana et al. (2010b). The vehicle configuration information model suggested in this PhD thesis explicitly emphasises the specific classes needed during vehicle configuration, which is a continuation of the *two-level* vehicle configuration by Haag (1998). The low-level configuration consists of the feature domain and the item domain. These two domains are also found in the vehicle configuration information model suggested in this PhD thesis. The *two-level* configuration is a characteristic of vehicle product families, where the number of alternative items is too high to be offered directly to customers. The two-level configuration model required more detailed classes in order to identify difficulties with current CR visualisation methods, see Paper C. The vehicle configuration information model suggested is therefore specific for vehicle configuration, but could be approached with the algorithms of a general problem, CSP. It was also necessary to create algorithms for the improved CR visualisation tool (RQ3), which could then beneficially be based on the algorithms for solving a CSP, for algorithms see Paper B and G. All computations, one of eight development process activities, could then be based on the algorithms for solving a CSP.

The second research question (RQ2) describes the vehicle CR development process, which will provide further examples of how the suggested vehicle configuration information model supports the description of the CR development process.

RQ2: How should vehicle configuration rules be developed time-efficiently and error-free?

A summary of the vehicle CR development process will now be described, for more details see Paper F. The three classes of vehicle configuration rules follow similar development processes, so there is no distinction necessary due to the configuration rule classes. The vehicle 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*.

Based on the process models from incremental knowledge engineering, the authoring process of configuration rules may be subdivided into the *elicitation*, *interpretation*, *formalisation* and *implementation* of configuration rules (Neubert, 1993). This PhD thesis has shown that the authoring process could be further described as an authoring method that has to be adopted, i.e. a specific systematic use of the logic operators NOT, IF-THEN, AND etc. The authoring methods found during the research work for this PhD thesis had two extremes: either the configuration rules were authored as *short* as possible, or as arbitrarily *long* as possible (up to hundreds of feature variants in a single configuration rule). Longer and shorter authoring of configuration rules are properties of single configuration rules. There is also an authoring method for a configuration rules set, i.e. two or more configuration rules. This authoring method for configuration rules sets is called *building blocks*, and the idea is to have a coherent usage of feature variants within the configuration rules set. The benefit of having a coherent usage of feature variants is a CR visualisation that is easier to overview. The authoring methods that were found were computable, i.e. authoring variations could be suggested from an initial authoring of configuration rules sets. This computability was crucial in order to implement the authoring methods into a CR visualisation. The authoring methods are described in detail in Papers B and G.

Based on the verification and validation techniques of knowledge-based systems, the evaluation process for configuration rules may be subdivided into *inspection*, *computation* and *testing* of configuration rules (Meseguer & Preece, 1995). The evaluation process includes several tools: a CR visualisation tool, a configurator and a virtual build tool. Among these tools, it is the CR visualisation tool that this PhD thesis has studied in detail. The configurator and the virtual build tool have been mentioned mainly in order to position research on product configuration.

A more time-efficient CR development process should avoid iterations, especially iterations where both product developers and product structure specialists are involved. Our approach has been to formalise the authoring methods and implement them in the CR visualisation tools in order to make these tools easier to use. With easier to use tools, fewer errors are made, and the product structure specialists and product developers are able to work more independently.

Literature on the CR development process has mainly focused on automated computations, e.g. Sinz et al (2003), Astesana et al. (2010b) and Preece et al. (1998). This PhD thesis also describes how computations may be automated, but the focus is on the inspection of vehicle configuration rules. The inspection has the aim of ensuring that the configuration rules are correct and complete. The level of efficiency during the inspection is dependent on both the authoring method adopted as well as the CR visualisation method. In this sense, the formalisation of the CR development process contributes to a development activity that previously lacked a formalised and computerised support.

RQ3: How can the CR visualisation tools be improved?

Little attention has been given in research to how to support the manual inspection of a knowledge-base by studying visualisation techniques (Baumeister & Freiberg, 2011). This PhD thesis describes research results for how to semi-automate the inspection of configuration rules by further developing the CRs that are used industrially. The improvement achieved is not only time-efficiency, but product developers capable of validating configuration rules, but not conducting the analysis because of heavy configuration rules computations, which could now be supported with the suggested CR visualisation tool.

First of all, there are automotive manufacturing companies that use text-based CR visualisation tools, e.g. lists or tables of vehicle configuration rules. These CR visualisation tools can be improved by instead applying a configuration rule matrix. As was stated in Nummelä (2006), configuration rules may be represented with a matrix consisting of symbols, e.g. “x”. This is a huge benefit for the overview when analysing configuration rules sets.

Secondly, one single matrix should visualise all classes of configuration rules simultaneously. This is to avoid misunderstandings that could arise from combining analysis from several classes of configuration rules. This has been achieved by introducing colour coding/shading in the demonstrator developed. At present, the industrial CR visualisation tools studied visualise each configuration rule class separately. According to the product developers, this caused difficulties as, for example, there were errors that could only be detected when more than one configuration rule class was inspected. There are matrix-based methods for configuration rules in the literature, e.g. the multi-domain matrix (MDM) called the K- and V-matrix by Bongulielmi et al. (2001). The K- and V-matrix has similarities to the industrial CR visualisation tools currently used, i.e. the fact that several matrices are used to represent vehicle configuration rules.

Included in the demonstrator of CR visualisation tool was also an implicit selection of feature variants, i.e. possibilities to author longer and shorter configuration rules were highlighted. Finally, fulfilment of typical requirements has also been highlighted, e.g. if there are any vehicle configurations that seem to have missing items.

5.2 Fulfilment of the research goal

Fulfilment of the research goal will now be discussed.

RG: To develop a systematic and efficient development process for vehicle configuration rules.

At the five automotive manufacturing companies that were involved in the research work for this PhD thesis, there was an even distribution among using lists, tables and matrices for CR visualisation. The CR visualisation method that was evaluated as the most user-friendly by the product developers was the matrix, for more details see Paper D. This PhD thesis has shown that it is possible to introduce a matrix-based CR visualisation method at the companies currently using lists or tables, see Paper E for more detailed research results. Interviewing the product developers at a company using a matrix-based CR visualisation method revealed that further improvement to the CR visualisation tool was to use one single matrix instead of a matrix for each class of configuration rule (product model authorisations, feature variant combinations rules, item usage rules). Validating the configuration rules, such as correctness and completeness of the configuration rules set, requires that all three classes of configuration rules are inspected. Also, authoring variations should be shown in order to avoid mistakes about implicit allowed/forbidden feature variants from the configuration rule set as a whole. A demonstrator with a new user interface for the matrix-based CR visualisation method was iteratively developed together with the product developers. The demonstrator was then used for usability tests in order to evaluate its benefit to the product developers. The usability test consisted of typical tasks during CR development, and the test results showed that the new CR visualisation method gave a more time-efficient and less error-prone development of configuration rules. Using the prototype, it was possible for a product developer to complete tasks in minutes that previously took hours/days/weeks. For more information about the usability tests and the demonstrator see Paper C.

Visualisation of vehicle configuration rules has not previously been studied in terms of improvements to user-friendliness. The example of a CR visualisation tool in Hami-Nobari & Blessing (2005), is used as a motivation that its user-friendliness is a problem, and Hami-Nobari & Blessing suggest the introduction of a new logic formalism in order to reduce the length and number of configuration rules. The commercial tools available, e.g. Tacton Studio (2013), visualise configuration rules, but do not address development and maintenance issues of vehicle configuration rules. Another finding that is aligned with the claim that CR visualisation is rarely studied comes from the literature review in Paper F. The paper states that the inspection is an activity that is rarely studied for Knowledge-based systems (KBS), of which the CR visualisation tools are an application.

5.3 Contribution

The contribution this PhD thesis makes concerns the research on vehicle configuration information model, on methods used when developing vehicle configuration rules, as well as the CR visualisation tools, see Table 5.1. As can be seen in the table, these areas are repeatedly addressed. For example, the CR visualisation methods have been addressed in Papers C, E and G. The most central contribution was the CR visualisation since:

- Improving the CR visualisation method was the approach to evaluate how to fulfil the PhD thesis research goal of a more efficient and user-friendly development of vehicle configuration rules. The difficulties of user-friendliness were also reported by Hami-Nobari & Blessing (2005). What has not previously been studied is how solely a modification in the CR visualisation method can make the tools more user-friendly.
- The systematic development process described was able to identify that the CR visualisation was central to the CR development process. The literature on the development of knowledge-based systems, e.g. MOKA (2000), had not previously been used for process analysis.

Table 5.1: Contribution of PhD thesis as a result of writing research papers A-G.

Contribution	Papers						
Vehicle configuration information model: Specification and identification of characteristics compared to other configuration information models	A			D			
CR authoring methods: Identification and formalisation.		B					G
CR visualisation methods: Currently used CR visualisation methods described, user needs identified and implemented with a usability-tested demonstrator.			C		E		G
CR development process: Specialised vehicle CR development process model.						F	

- The CR authoring methods were formalised in order to be implemented in the CR visualisation. Different CR authoring is an example of code refactoring, i.e. the code is changed but not its external behaviour. The formal methods in literature have previously analysed the external behaviour (Astesana et al., 2010; Sinz et al., 2003).
- When studying the vehicle configuration information model, it was found that the main challenge was to develop the three classes of vehicle configuration rules in parallel, which has been addressed with the proposed matrix-based CR visualisation. The use of multiple matrixes for variant-rich products has also been identified by Bongulielmi et al. (2001). This thesis shows how these matrices can be merged into one single matrix.

5.4 Validation of research results

The following will provide some examples of how the validation threats, see Chapter 3.4: Validation approach, were addressed:

- **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.* The studies for Papers A, D and E took place as long-term involvement with the companies, in total 1.5 years.

- **Respondent validation:** *Systematically collecting feedback about the data and the conclusions from the people that are studied.* Paper C included an interview study, where the interview protocols and the conclusions were sent back to the interviewees in order to receive feedback. Also, throughout the research project, especially during the first half (Paper A-C), a group of experts from the automotive industry reviewed the research results.
- **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.* All papers that describe case studies (Papers A, B, C, D, E and F) used some type of triangulation. Most commonly, documents were studied as a complement to interviewing and/or observing.
- **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 provides support for qualitative claims as well as assessing the amount of evidence in the data.* Time estimations, even if approximate, were collected in Paper A and Paper C.

Another piece of supporting evidence for the validity of the research results is the interest experienced from the automotive manufacturing industry. The research work started at Company A was extended to also include Companies B and C at their request. After the publication of the Licentiate thesis half way through the PhD, it was possible to establish an internal collaboration with Companies D and E. Buur (1990) calls this is *verification by acceptance*.

5.5 Generalisability and transferability

Qualitative researchers emphasise movements from observations to descriptions, and then finally to theory generation in order to achieve generalisations (Johnson & Christensen, 2012). The theory explains how something operates in general and moves away from one single research study. Transferability is the possibility of making connections between the research results and other contexts. Generalisability and transferability are therefore not mutually exclusive, and can be discussed in parallel. The research results will now be discussed in terms of the aspects generalisability and transferability, divided into the subsections of configuration information model, CR development process and CR visualisation methods.

5.5.1 Configuration information model

What the vehicle configuration information model suggested in this PhD thesis describes is the fact that there are several classes of constraints (feature variant combination rules, product model authorisations, item usage rules), and several classes of variables and variable values (feature families, feature variants, product models, items). Vehicle configuration can be viewed as a special case of the general problem called Constraint Satisfaction Problem (CSP). With CSP algorithms as a solution approach, the vehicle configuration information model has to consist of variables, variable values and constraints. Any configuration can be expressed as a CSP, which means that non-automotive companies can also model their product configuration as a CSP.

The vehicle configuration information model suggested, along with the problems identified, has been verified at the five automotive manufacturing companies studied in this PhD thesis. This thesis' research results are to a large extent based on the problems that occur when developing configuration rules for two domains and not one. Apart from vehicle product families, there are other product families that potentially have a configuration information model with two domains. A possible example would be houses/buildings, which follow models allowed a certain product variety and additional equipment, such as flooring materials of *oak* and *birch*, kitchen cupboards *Mono or White Square* etc. (Willa Nordic, 2013). As an example, an IKEA kitchen cupboard exists in three feature families: (1) colour of the frame, (2) model and colour of the front, and (3) size. The three feature families have 26 feature variants and there are 126 items.

5.5.2 CR development process

The process models for the CR development process will now be discussed in terms of generalisation and transferability. Before discussing details of the process, it can initially be concluded that the CR development process is relevant for all three classes of vehicle configuration rules, see Fig. 5.1. As can be seen in the figure, there are three similar development processes that can be conducted in parallel for the three classes of vehicle configuration rules. This is a specific property of the vehicle CR development process, and a consequence of the vehicle configuration information model. It may however be assumed that there are other products, e.g. modular

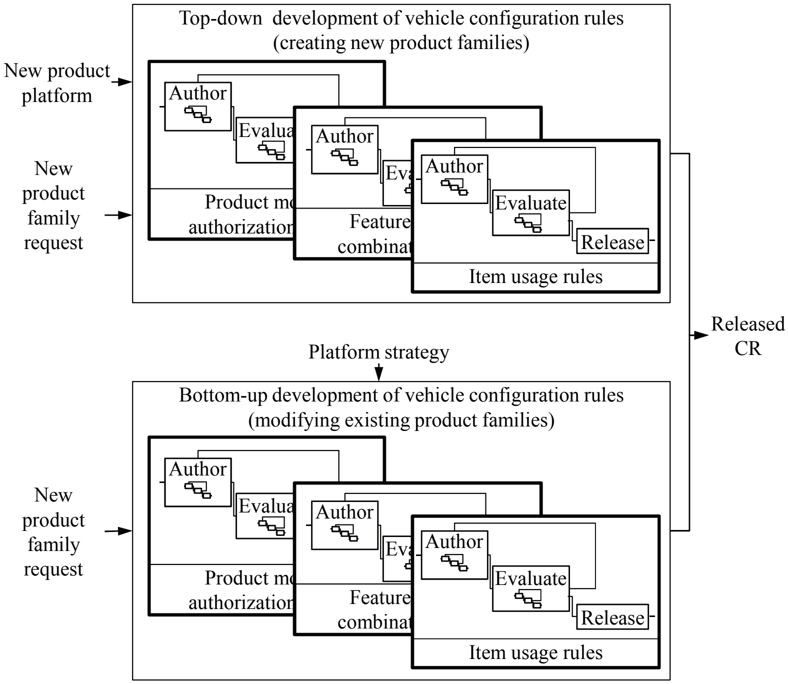


Figure 5.1: Process overview of vehicle CR development.

houses (Willa Nordic, 2013), with several classes of configuration rules, which make the findings transferable.

The possibility of reformulating logic expressions is a general characteristic, and not specific for vehicle configuration rules. A requirement for the authoring methods is, however, that the configuration rules are expressed as logic expressions. There is therefore no transferability of the authoring methods to products that are configured using parametric design rules, which use the product geometry.

The more detailed process models that were created for the CR development process were based on KBS development frameworks. The framework as a basis for the process models are therefore not specific to only vehicle configuration rules but for development of knowledge in general. The development of CRs was thus decomposed into authoring and evaluation of configuration rules. The controls and mechanisms in the process models (arrows from below and from above to the process activities) are the focus of attention when discussing generalisation and transferability.

5.5.3 CR visualisation

The CR visualisation methods that have been studied further in this PhD thesis are lists, tables and matrices. This thesis has shown that the CRs formulated as logic expressions can be visualised using either one or the other method, see Paper E. The research results therefore hold for any logic-based configuration rules, not only vehicle configuration rules. A matrix, however, requires the configuration rules to be logic expressions, otherwise they are difficult to be visualised as a matrix. This thesis argues that there are benefits for user-friendliness when using a matrix-based CR visualisation method. This recommendation holds for not only for vehicle configuration rules, but any configuration rules that could be visualised with a matrix.

5.6 Summary

This chapter has discussed the research goal of a more efficient CR development process, and has also discussed answers to the research questions. The contributions were then listed with a subsequent discussion concerning validity, generalisation and transferability.

6 Conclusions

This PhD thesis has approached the goal of finding a less error-prone and more time-efficient vehicle CR development process from scratch by studying the vehicle configuration information model, the vehicle CR development process and the vehicle CR visualisation.

The key findings summarised are the specification of the vehicle configuration information model, which is a more elaborated version of the two-level configuration described in Haag (1998), but could nevertheless be approached with algorithms solving the constraint satisfaction problem (CSP). This vehicle configuration information model is thereby based on the feature and the item domains. The two domains, along with the three classes of vehicle configuration rules, challenge both the CR development process as well as the CR visualisation tools. A systematic process modelling and analysis showed that the process activity constituting the decision point for iterations in the CR development process is the inspection of configuration rules. This gives the CR visualisation tools a central role in an efficient and error-free CR development process. A demonstrator for an improved CR visualisation method was created and evaluated with the product developers. Test results showed that it was possible to achieve greater user-friendliness as well as a higher degree of automation with the demonstrator compared to the CR visualisation tools currently used. The automation mainly consisted of computations for combining all three classes of vehicle configuration rules into one single CR visualisation. It was also possible to detect variations in the CR authoring methods and show them directly in the CR visualisation. Usability tests showed that typical CR development tasks which were taking hours/days/weeks with the CR visualisation tools currently used, could be done in minutes with the demonstrator.

The claims about the findings will now be explained in more detail and commented upon. The claims are divided into sections: vehicle configuration information model, vehicle CR development process and vehicle CR visualisation tools:

- **Vehicle configuration information model:** Configuration information models can be based on either one or two domains, corresponding to what this PhD thesis calls the feature and item domains. The difficulties identified during the CR development process, as well as when creating the CR visualisation, originate from the presence of two domains in the vehicle configuration information model. The single-domain configuration information model only contains intra-domain configuration rules, called *product model authorisations* and *feature variant combination rules*. The vehicle configuration information model, as well as in theory all two-domain configuration information models, also contains inter-domain configuration rules. These configuration rules are called *item usage rules* in this PhD thesis. Furthermore, the vehicle configuration rules are logic expressions, i.e. they contain logic operators such as AND, NOT, IF-THEN etc. Configuration rules based on logic have drastically different properties compared to configuration rules expressed using parametric design rules. Design rules are mathematical equations capable of modifying the design of items. Vehicle configuration relies entirely on predesigned items; hence logic expressions that only control the compatibility between items can be used. The logic-based vehicle configuration rules will be of central importance when discussing the development process for CRs and how the CR visualisation is created. For example, the possibilities when creating the CR visualisation with logic-based configuration rules goes far beyond the possibilities with parametric design rules. The next two bullet points will elaborate the claims for the CR development process and the CR visualisation tools.

- **Vehicle CR development processes:** The starting-point for the research work on the vehicle CR development process was an informal process which was often dependent on teamwork between several product developers. By referring to the CR development process as a knowledge acquisition and knowledge evaluation process, see Paper F, it was possible to systematically approach the product configuration literature. The knowledge acquisitions, called CR authoring, consist of the four process activities: elicitate, interpret, formalise and implement (Neubert, 1993). The knowledge evaluation, called the CR evaluation, consisted of the three process activities: inspection, computations and testing (Meseguer & Preece, 1995). Through this systematic modelling approach, it was possible to conclude that the inspection is a central decision point for the CR development process. The inspection is done with the CR visualisation tools, which could be improved in order to make the inspection more time-efficient and easier to conduct. The inspections are done manually, but an increased time-efficiency could be achieved by automating some typical checks during the CR development. For example, the discussion concerning the CR authoring methods could be avoided with a CR visualisation method including authoring method variations. The CR development methods were mainly manual computations, and the expertise required to conduct them was mainly found among the configuration rules specialists. Computations for increasing automation have been a major driver for product configuration research, e.g. (Meseguer & Preece, 1995) and (Sinz et al., 2003). The computations that are described in the literature have the aim of complete automation or do not show how the computation results should be presented to the product developer. This exclusion of the product developer is a major deficiency, as the main goal is to have a valid set of configuration rules, i.e. complete and correct configuration rules. It is currently only possible for product developers to conduct validation of configuration rules.

- **Vehicle CR visualisation tools:** The industrial CR visualisation tools are based on lists, tables or matrices. Half of the automotive manufacturing companies studied use a matrix-based CR visualisation method, while the other half have a list or a table. Independently of the initial CR visualisation method, it is generally feasible to apply a matrix-based CR visualisation method instead. The matrix-based CR visualisation method has been found to have the advantage of increasing ease of use for product developers. Further improvements were achieved at an automotive manufacturing company currently using a matrix-based CR visualisation method, that was possible to test with a demonstrator. The improved CR visualisation method was easier to use and supported the CR development process better by (1) including the visualisation of authoring methods, and (2) visualising all three classes of configuration rules in one single CR visualisation. Evaluations together with real users of the industrial CR visualisation tools have shown that the new CR visualisation 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 PhD thesis offers two interesting future research opportunities on development processes for vehicle configuration rules: the product rationalisation process and the process for making forbidden vehicle configurations allowed.

7.1 Product rationalisation study

This thesis has studied the development of vehicle configuration rules by looking at the development process. This process is, however, most commonly an extension or substitution of feature variants and items within the current product families. Continuing to introduce more feature variants and items may potentially result in the need to actually reduce the size of a product family. What this thesis has not studied empirically is how this is achieved today.

Another process that has not received a great deal of attention is the analysis of forbidden vehicle configuration rules, which will be discussed in the next section.

7.2 Analysing vehicle configurations that were previously forbidden

When developing vehicle configuration rules, the time spent is motivated by the new product features that will be possible to offer customers. Not all feature variant combinations are developed for the new product features. This is due to financial reasons, as well as technical feasibility. When customers request vehicle configurations that are not allowed, a study could be undertaken to show what development costs may be accepted. This study is thus limited to vehicle configuration alone. If there are more customers requesting a certain feature variant combination that is not allowed according to the vehicle configuration rules, it can be discussed whether the configuration rules should be reformulated. This process then needs to study in detail why vehicle configuration rules have been authored, as well as the effort to reformulate them. It should be possible to master this process efficiently as is this case with all other processes, and one of the industrial collaborators has recently been developing a new information system support for this process. It would have been interesting to study this new support in order to further understand the details of the vehicle CR process.

References

Akao, Y. (1994), “Development History of Quality Function Deployment”, *Asian Productivity Organization,* pp. 339.

Akerkar, R.A. & Sajja, P.S. (2010), “Knowledge-based systems”, *Jones and Bartlett Publishers,* Mississauga, Canada.

Andreasen, M. M. (1992), “Designing on a Designer’s Workbench (DWB)”, *Proceedings of the 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. (2010a), “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.

Astesana, J.M., Cosserat, L. & Fargier, H. (2010b), “Constraint-based Vehicle Configuration: A Case Study”, *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI2010),* IEEE, pp. 68–75.

Ayel, M. & Laurent, J-P. (1991), “Verification, Validation and Test of Knowledge-based Systems”, *John Wiley & Sons,* New York.

Schuh, G. & Kampker, A. (2010), “Variant Management”, lecture in Product Management at RWTH Aachen.

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.

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

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 01)*, pp. 571-578.

Bongulielmi, L., Henseler, P.C., Puls, C. & Meier, M. (2002), "The K- & V-Matrix-Method in Comparison with other Matrix-based Methods supporting Modular Product Family Architectures", In Sigurjonsson, J.B. (eds.), *Proceedings of NordDesign*, Trondheim.

Buur, J. (1990), "A Theoretical Approach to Mechatronics Designs", *PhD thesis*, Technical University of Denmark, Lyngby, Denmark.

Burr, H., Müller, M. & Vielhaber, M. (2007), "EIMS - A Framework for Engineering Process Analysis", In *Proceedings of International Conference on Engineering Design (ICED07)*, paper 428.

Bühne, S., Lauenroth, K. & Pohl, K. (2004), "Why is it not Sufficient to Model Requirements Variability with Feature Models?", *Proceedings of Automotive Requirements Engineering (AURE04)*, pp. 5-12.

Büning, H.K. & Lettmann, T. (1999), "Propositional Logic: Deduction and Algorithms", *Cambridge University Press*, Cambridge, 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.*

Czarneck, K., She, S. & Wasowski, A. (2008), "Sample Spaces and Feature Models: There and Back Again", *Software Product Line Conference (SPLC'08),* Limerick, 8-12 September, pp. 22-31.

Darr, T., McGuinness, D. & Klein, M. (1998), "Special Issue on Configuration Design". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol. 12, No. 4.

Dassault (2014), "Enovia X-BOM Cost Analytics",
http://www.3ds.com/uploads/tx_3dsportfolio/2012-11-23-enovia-cst.pdf [accessed 2014-02-24]

Di Cosmo, R. (2006), "Report on Formal Management of Software Dependencies", *Environment for the Development and Distribution of Open Source Software*, FP6-IST004312.

Domazet, D.S., Yan, M.C., Calvin, C.F.Y., Kong, H.P.H. & Goh, A. (2000) "An Infrastructure for Inter-organizational Collaborative Product Development", *International Conference on System Sciences - HICSS,* Hawaii, University of Hawaii/Shidler College of Business.

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.

Edwards, K. & Pedersen Lindgaard, J. (2004), "Product configuration systems – implications for product innovation and development", *Proceedings from the international conference on Economic, Technical and Organisational aspects*, Institut for Produktion og Ledelse, DTU.

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

Estublier, J., Leblang, D., van der Hook, A., Conradi, R., Clemm, G., Tichy, W. & Wiborg-Weber, D. (2005), "Impact of Software Engineering Research on the Practice of Software Configuration Management", *ACM Transactions on Software Engineering and Methodology,* Vol. 14, No. 4, pp. 1-48.

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

Faltings, B. & Freuder, E.C. (1998), "Guest editors' introduction: Configuration", *IEEE intelligent systems & their applications*, Vol. 13, No. 4.

Fang, D., Hammer, J. & McLedo, D. (1991), "The Identification and Resolution of Semantic Heterogeneity in Multidatabase Systems". In Kambayashi, Y., Rusinkiewicz, M. and Sheth, A. (eds.) *International workshop on Interoperability in Multidatabase Systems (IMS'91),* Kyoto.

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

Forza, C. & Salvador, F. (2002), "Managing for variety in the order acquisition and fulfilment process: the contribution of product configuration systems", *International Journal of Production Economics,* Vol. 76, pp. 87-98.

Franke, N. (1998), "Configuration research and commercial solutions", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol. 12, No. 4, pp. 295-300.

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.

Gupta, U.G. (1991), "Validating and verifying knowledge-based systems", *IEEE Computer Society Press*, Los Alamitas, 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.

Günther, A. & Hotz, L. (1999), "KONWERK – a domain independent configuration tool", *Proceedings from AAAI Workshop on Configuration,* AAAI Press, Orlando, FL, USA, pp. 10-19.

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

Hall, A. (1990), “Seven myths of formal methods”, *IEEE Software,* Vol. 7, No. 5, pp. 11-19.

Hami-Nobari, S. & Blessing, L. (2005), “Effect-oriented Description of Variant Rich Products”, *Proceedings of the 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.

Heinrich, M. & Jüngst, E. (1991), “A resource-based paradigm for the configuring of technical systems from modular components”, *Proceedings of IEEE Conference on AI Applications (CAIA)*, pp. 257–264.

Hirel, G. & Hug, H. (2009), “Harmonized Exchange of Bill of Material Data for the Next Truck Generation”, *Integrated Virtual Product Creation Symposium - iViP*, Berlin, ProSTEP.

Hitchins, D.K. (2003), “Advanced Systems – Thinking, Engineering, and Management”, *Artech House Inc.*, Norwood, MA, USA.

Hollmann, O., Wagner, T. & Günther, A. (2000), “EngCon: A flexible domain-independent configuration engine”, *Proceedings of Configuration (ECAI 2000- Workshop)*, pp. 94-96.

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.

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 (1994), ISO 10303-1:1994, *International Organization for Standardization,* http://www.iso.org [accessed 2013-12-13].

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

Jacobson, I. & Lindström, F. (1991), "Reengineering of Old Systems to an Object-oriented Architecture", *ACM SIGPLAN Notices,* Vol. 26, No. 11, pp. 340-350.

Jiao, J., Simpson T.W. & Siddique, Z. (2007), ''Product Family Design and Platform-Based Product Development: a State-of-the-Art Review'', *Journal of Intelligent Manufacturing*, Vol. 18, No. 1, pp. 5-29.

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.

Johnson, B. & Christensen, L. (2012), "Educational Research", *Sage Publications Inc.,* Thousands Oaks, CA, USA.

Junker, U. (2006), "Configuration", In Rossi, F. et al (Eds.) Handbook of Constraint Programming, Foundations of Artificial Intelligence, *Elsevier Science*, pp. 837-874.

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.

Kahn, K.B. (2012), "PDMA handbook of new product development", *Johan Wiley & Sons,* Hoboken, New Jersey, USA.

Krebs, T. (2009), "A Knowledge Management Framework That Supports Evolution of Configurable Products", *PhD Thesis,* Rhombos-Verlag, Berlin.

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 the Conference on Graphics and Visualization in Engineering*, Clearwater.

Leonard-Barton, D. (1998), "The case for integrative innovation and expert system at digital", *Sloan Management Review*, Vol. 29, No. 1, pp. 7-19.

Levandowski, C. (2012), "Towards Development Platforms", *Licentiate Thesis*, Chalmers University of Technology, Gothenburg, Sweden.

Lindemann, U., Maurer, M. & Braun, T. (2009), "Structural Complexity Management", *Springer-Verlag,* Berlin Heidelberg.

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

Loomeo (2011), “LOOMEO – Graph“, http://youtu.be/8kyLPNHDZUw accessed [2014-02-17]

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.

MacDuffee, C.C. (1940), “Introduction to Abstract Algebra”, *Wiley,* New York.

Malmqvist, J. (2002), “A Classification of Matrix-based Methods for Product-modelling”, *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.

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

Mackinlay, J. (1986), “Automating the design of graphical presentations of relational information”, *ACM Transactions on Graphics*, Vol. 5, No. 2, pp. 110-141.

Markson, H. (2007), “Achieving CAD Interoperability in Global Product Design Environments”, White Paper, *SpaceClaim Corporation.*

McKay, A., Erens, F.J., & Bloop, 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. (2004), “On the Development and Sales-delivery Process of Configurable Products”, *Licentiate Thesis,* Chalmers University of Technology, Gothenburg, Sweden.

Mesihovic, S. & Malmqvist, J. (2004), “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.

Meyer, M.H. & Lehnerd, A.P. (1997), “The Power of Product Platforms: Building Value and Cost Leadership”, *Free Press,* New York, USA.

Mittal, S. & Frayman, F. (1989), “Towards a Generic Model of Configuration Tasks”, *Proceedings of 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].

Männistö, T., Soininen, T. & Sulonen, R. (2001), "Product Configuration View to Software Product Families", *Proceedings of Software Configuration Management Workshop (SCM-10).* pp. 14-15.

Naiman, C.E. & Ouksel, A.M. (1995), "A Classification of Semantic Conflicts in Heterogeneous Database Systems", *Journal of Organizational Computing*, Vol. 5, No. 2, pp. 167-193.

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

Nilsson, N.J. (1980), "Principles of Artificial Intelligence", *Morgan Kaufmann,* San Francisco.

Nobelius, D. (2002), "Managing R&D Processes-Focusing on Technology Development, Product Development, and their Interplay", *Ph.D. Thesis*, Operations Management and Work Organization, Chalmers University of Technology, Gothenburg, Sweden

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]

Nummelä, J. (2006), "Integrated Configuration Knowledge Management by Configuration Matrices", *PhD thesis*, Tampere University of Technology.

O'Donnell, J., Hall, C. & Page, R. (2006), "Discrete Mathematics Using a Computer", *Springer-Verlag*, London.

Ohl, S. (2000), "Prognose und Planung variantenreicher Produkte am Beispiel der Automobilindustrie", [Prognoses and planning for variant-rich products with an example from the automotive industry],] *VDI-Verlag,* Düsseldorf.

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

Oracle (2014), "Agile Product Cost Management", http://www.oracle.com/us/products/applications/agile/product-lifecycle-management/product-cost-management/ [accessed 2014-02-24]

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

Peltonen, H. (2000), "Concepts and an Implementation for Product Data Management", *PhD thesis,* Helsinki University of Technology.

Penoyer, J.A., Burnett, G.J.F.D. & Liou, S.-Y. (2000), "Knowledge-based product lifecycle systems: principles of integration of KBE and C3P", *Computer-Aided Design,* Vol. 32, No. 5-6, pp. 311-320.

Pinfold, M. & Chapman, C. (1999), "The application of knowledge-based Engineering Techniques to the Finite Element Mesh Generation of an Automotive Body-in-white Structure", *Journal of Engineering Design*, Vol. 10, No. 4, pp. 365-377.

Pratt, M.J. (2001) "Introduction to ISO 10303 - the STEP Standard for Product Data Exchange", *Journal of Computing and Information Science in Engineering*, Vol. 1, No. 1, pp. 100-101.

Preece, A., Grossner, C., Chander, P.G. & Radhakrishnan, T. (1998), "Structure-Based Validation of Rule-Based Systems", *Data and Knowledge Engineering,* Vol. 26, pp. 161-189.

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

PTC (2014), "PTC Windchill Cost", http://www.ptc.com/product/windchill/cost [accessed 2014-02-24]

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

Russell, S. & Norvig, P. (2009), "Artificial Intelligence: A Modern Approach", Chapter 5 – Constraint Satisfaction Problems*, Prentice Hall*, Upper Saddle River, USA.

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.

Schuh, G. & Jonas, I. (1997), "Variantenreduzierung im Verbund – Praktikable Methode zum Variantenmanagement: Ein Leitfaden zur Beherrschung der Variantenvielfalt". [Variant reduction in the Association – Practical Methods to Variant Management: A Guide to Control the Variant Variety], *Verbundinitiative Automobil Nordrhein-Westfalen*, Düsseldorf, Germany.

Schuh Group, (2014), "Complexity Manager", www.complexitymanager.com accessed [2014-12-17]

Shapiro, A.R. (2006), ''Measuring Innovation: Beyond Revenue From New Products'', *Research-Technology Management,* Vol. 49, No. 6, pp. 42–51.

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.

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.

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 (2013), “Design Automation with Tacton”, http:// http://www.tacton.com/design-automation/design-automation-with-tacton/ [accessed 2013-08-29].

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

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

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.

Turesson, B. (2006), “PLM from a holistic Business Perspective”, *Proceedings of Nordic Conference on Product Lifecycle Management (NordPLM’06),* Göteborg, Sweden.

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. (2003), “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.

Voronov, A., Åkesson, K., & Ekstedt, F. (2011), ”Enumeration of valid partial configurations”, In *Proceedings of Workshop on Configuration (IJCAI 2011),* Vol. 755, pp. 25-31.

Waldinger, R.J. & Stickel, M.E. (1992), “Proving properties of rule-based systems”, *International Journal of Software Engineering and Knowledge Engineering,* Vol. 2, No.1, pp. 121-144.

Weber, J. (2009), “Automotive Development Processes – Processes for Successful Customer Oriented Vehicle Development”, *Springer-Verlag*, Berlin Heidelberg.

Wheelwright, S.C. & Clark, K.B. (1992), "Creating project plans to focus product development", *Harvard Business Review*, Vol. 70, No. 2, pp. 67–83.

Willa Nordic (2013), "Hus med attityd – utrustning och tillbehör", Retrieved from www.willanordic.se at 2013-11-25.

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

Zeuthen, H.G. (1870), "Sur les Points Fondamentaux de Deux Surfaces Dont les Points se Correspondent Un à Un", *Comptes Rendus des Séances de l'Académie des Sciences,* Vol. 7, p. 742.

Zimmerman, T. (2008), "Implementing PLM across organisations – for multi-disciplinary and cross-functional product development", PhD thesis, Chalmers University of Technology, Gothenburg, Sweden.

Appended Papers A-G

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.

Paper D Tidstam, A. & Malmqvist, J. (2013), "Obstacles and development of support for translation of configuration rules", *Proceedings of the International Conference on Engineering Design (ICED13),* Seoul, Korea, 19-22 July.

Paper E Tidstam, A. & Malmqvist, J. (2013), "Comparison of Configuration Rule Visualization Methods", *Proceedings of Product Lifecycle Management (PLM13),* Nantes, France, 6-10 July.

Paper F Tidstam, A. & Malmqvist, J. (2013), "A systematic process for developing vehicle configuration rules", *Submitted to Scientific Journal.*

Paper G Voronov, A., Åkesson, K., Tidstam, A., Malmqvist, J. & Fabian, M. (2014), "Formal methods for the inspection of vehicle configuration rules", *Submitted to Scientific Journal.*