TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 589 • Publication 589

Juho Nummela

Integrated Configuration Knowledge Management by Configuration Matrices – A Framework for Representing Configuration Knowledge

Tampere 2006

Tampereen teknillinen yliopisto. Julkaisu 589
Tampere University of Technology. Publication 589

Juho Nummela

Integrated Configuration Knowledge Management by Configuration Matrices – A Framework for Representing Configuration Knowledge

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Konetalo Building, Auditorium K1702, at Tampere University of Technology, on the 24th of February 2006, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2006

ISBN 952-15-1564-3 (printed)
ISBN 978-952-15-2156-0 (PDF)
ISSN 1459-2045

ABSTRACT

To satisfy customer needs a product family has to offer distinctive features while the production struggles with the large variety of different modules and products that can be produced. Configurable products provide the ability to offer different variants for customers. Configurability is often established by modular product family architectures. Modularity and configurability are important factors for producible product families. Producibility offers means to provide effective product family architectures to support the organization designing and manufacturing configurable products. Producibility includes the idea that product family structures need to be simultaneously configurable and support the production system. This has implications to the types of modules used in the product structures. Configurable products involve large amounts of information related to dependencies between different elements of the product structure. The representation of configuration knowledge for configurable products is important. This knowledge needs to be visually available in order to provide efficient basis for the configuration process to be effective. While configuration knowledge needs to be documented to be used as a basis for a configurator, the generic product structure provided by the configuration models establishes a basis for a variety of tools that can be used to support the organization using configurable products as a basis for satisfying the customer needs.

The problem for configurable products is that the information related to the products needs to be easily presented to support the organization. This knowledge is usually tacit. While configuration knowledge is not visual also modularity and configurability are often misunderstood in industrial environments while producibility is not even considered. Even if configurable products are utilized, the use of a configurator can be impossible due to the changes that take place during the lifecycle of a product family. Even the implementation of a configurator can be impossible due to the problems in modeling and documenting configuration knowledge. Also the changes during the lifecycle of customer orders mixed with the changes to the configuration models have many effects to the organization using configurable products.

The main emphasis of this research is to provide understanding of producibility in the context of configurability and modularity. In order to efficiently use the configuration knowledge to support the organization, configuration matrices are established for documenting this knowledge. The main purpose for the notation used in representing the configuration knowledge is to be as simple as possible so that real benefits for industrial environments can be achieved. The configuration matrices are used to present the modular systems related to the configurable products, thus the configuration knowledge is visualized through these matrices. The configuration matrices also provide the generic product structures for the modular systems. Based on the configuration matrices a framework for configuration knowledge is established in the form of different types of tools that can be used to support the organization. This framework also includes a configurator that can be used in changing environments while the configuration knowledge needed for the configurator is automatically derived from the configuration matrices. The tools presented in the framework integrate different parts of the organization in the context of configuration knowledge. An in-depth case study is discussed to validate the results provided by this research.

ACKNOWLEDGEMENTS

This research is the result of cooperation between many disciplines. First of all the interest of the case company's plant director Heikki Ojala concerning this field of research enabled firstly the Master thesis and finally this Doctoral thesis to be accomplished. The close co-operation between the Institute of Production Engineering of TUT (Tampere University of Technology) and the case company enabled this research by locating the researcher to the industrial environment. Thank you for the opportunity and support.

Professor Torvinen, from TUT has been the source of inspiration for this research from the beginning. As being my mentor and close supporter, the high rate of accomplishments and the enthusiastic research attitude could be maintained during this research. Professor Torvinen has also enabled this research by taking it as a part of the Process-MSDD (Manufacturing System Design and Decomposition) project funded by TEKES (The National Technology Agency of Finland). Emeritus professor Lapinleimu and Doctor Bongulielmi have also contributed to this research by the conversations conducted during this research. Finally thanks go to colleagues in TUT for good conversations and suggestions made during conference trips and seminars as well as meetings.

From the case company I would like to thank Timo Karhunen, Markku Penttinen and the fellows working with configuration knowledge and module interface maintenance for the support and new ideas that started to surface after the implementation.

Finally, one of the biggest issues is the family and good relationships that made it all happen. For all the sacrifices and understanding that this research has required from my closest ones, I am grateful for this flexibility. Without your support this would not have been possible. Thank you Kaisa and "The Nums".

Vieremä, February 8th, 2006

Juho Nummela

ABBREVIATIONS AND TERMINOLOGY

ASP	Active Server Pages
BOM	Bills of Material
CCT	Configurability Conformance Test
Configurability	A property of the modular system that enables the establishment of customer specific product individuals
Configurable product	Product based on a modular system that provides different variants of the product to the product family
Configuration knowledge	All knowledge related to configurable products that forms the basis for configuration models
Configuration matrix	A square matrix presentation that provides means to document, model and maintain configuration knowledge efficiently
Configuration model	Model that documents the configuration knowledge related to configurable products
Configuration rules	Rules that determine dependencies between options as well as between options and modules
Configuration task	A process that uses a configurator or manual selection of elements to establish a customer specific product individual
Configurator	A software package that uses configuration knowledge to establish customer specific product individuals
CSD	Collective System Design. A framework to holistically consider product development and manufacturing systems simultaneously.
DFMA	Design for Manufacturing and Assembly
DfV	Design for Variety
DSM	Design Structure Matrix
Dymo	Dynamic modularization
ERP	Enterprise Resource Planning
Excel	Microsoft Excel spreadsheet software
Functional module	An entity of a product that cannot necessarily be assembled. Executes a function or a set of functions alone of accordance with other modules
Generic product structure	All the modules based on the modular system that can be included into the configured product individual according to the configuration knowledge
MBI	Modularization by Integration
MFD	Modular Function Deployment
MIM	Module indication matrix
MIT	Massachusetts Institute of Technology
Modular system	Group of modules that form the generic product structure and the basis for configuration models and product families
Modularity	A property of a system that is related to its structure and functionality
Modularization	Task of decomposing the product architecture into modules

Module	A structural entity of a final product architecture that has well defined interfaces
MPA	Modular Product Architecture
MRP	Material Requirements Planning
MSDD	Manufacturing System Design and Decomposition. Part of CSD model
MySQL	Database application
ODBC	Open DataBase Connectivity
PDM	Product Data Management
P-DSM	Product modeling Design Structure Matrix
Ponsse	The case company
PSD	Production System Design Laboratory
Process-MSDD	Project to further develop and validate MSDD systematics through case studies in Finnish industry
Producibility	Property of a product that supports the production system. Important issue supporting CSD model.
Product architecture	Decomposition of functional entities into physical building blocks of a product
Product family	Collection of different product individuals established according to the configuration knowledge related to configurable products. All variants derived from the modular system
Product platform	A group of subsystems and interfaces that form a general structure
QFD	Quality Function Deployment
SQL	Structured Query Language
Structural module	Structural entity that has been defined considering the production system
TEKES	The National Technology Agency of Finland
TUT	Tampere University of Technology
VB	Visual Basic programming language
VDI	Verein Deutsche Ingenieure
WIP	Work In Progress
5S	Sustain, Standardize, Sort, Set in order, Shine. Japanese philosophy to support standardized work. Part of CSD model.

LIST OF FIGURES AND TABLES

Figures

Tables

CONTENTS

1. INTRODUCTION

Customers consider product variation important while companies are struggling with the different product variants needed to be produced. The problem is that differing customer needs require freedom of choice in products which produces deviation to the production system. Thus, not all the products that pass the production are similar between each other. Next to the increasing number of different product variants, the time is also often limited to deliver these new products, product families or features. It can be easily noted that all the main departments of a manufacturing company are involved when considering customer specific products and the processes related. There is a clear conflict between the wants of the customers and the emphasis of a manufacturing company to be as cost-effective as possible.

Configurable products are one way of satisfying differing customer needs. Tiihonen and Soininen (1997) suggest that configurable products provide means to offer variety for customers through customer specific adaptation. Configurability of a product is related to the product structure and can be considered to be the product architecture's ability to provide different customer specific product individuals, i.e. products based on customer specifications. Modularity is often offered to solve the problem of configurability. Modularity can be seen as a property of the system that is related to its architecture and functionality (Ulrich and Eppinger 2000). Configurability and modularity are factors of producibility. Producibility can be seen as the property of a product that supports the production system (Lapinleimu 2000). Configurability on the other hand is a property of a product that supports the configuration process. Tiihonen and Soininen (1997) argue that the goal of configurable products is to provide means to resolve the conflict mentioned above. For configurable products generic product structure includes all the possible modules from the modular system that provides the means for producible product structures.

For configurable products, product family is defined as the sum of different product variants that can be derived from the modular system (Pulkkinen et al. 2004). This modular system includes all the necessary information (e.g. modules, options, platforms, dependencies) that is needed to provide competitive customer specific product individuals. This information can be develop into reusable knowledge by configuration models (see definition in 2.3.1) and defined as configuration knowledge. This thesis is considering producibility of the product families in the context of configurability and modularity. Thus, the aim is to provide configurable products while simultaneously enabling the production system to provide these product variants efficiently.

Modularity needs to be considered holistically (Holmqvist and Persson 2004) considering all the related stakeholders. Sub-optimizing the product architecture can be easily done when considering the benefits only locally. Configurability and modularity are both important factors for producibility. Even if modularity simplifies the product architectures greatly, the configuration knowledge related to configurable products is very central to the company offering intense product variation. Documenting this knowledge can be time consuming and is often considered tacit knowledge as presented by Forza and Salvador (2002a) and Peltonen et al. (1998). This is also considered to be one of the main obstacles when implementing a configurator. Thus, the configuration knowledge is central while its documentation needs to be systematically handled in order to provide quick processes and deliver the benefits related to configurable products.

Even if the documentation of configuration knowledge can be tedious, the reality is that while different changes take place during the lifecycle of the product family as well as during the lifecycle of a customer order, the importance of configuration knowledge and its management grow substantially. Configurator is a software package that uses configuration knowledge to provide customer specific product individuals automatically from the customer specifications. Thus, a configurator enables the use of customer specific adaptation of a product (Tiihonen and Soininen 1997). If the configuration knowledge is not maintained in efficient ways, the configurator can also be impossible to maintain. Usually configurable products experience changes while the capability of product design to provide modularity determines the implications of these changes to other parts of the organization.

It can be noted that configurability and modularity are heavily connected while the processes of product design largely determine the way the rest of the organization can operate in the context of product architecture. This thesis provides a framework built around configuration knowledge that provides insights how configuration knowledge affects the entire organization. One of the goals for configurable products is to use a configurator during the order-delivery process. The framework also includes a model of a configurator that provides means to establish customer specific product individuals in changing environments. Thus, the configuration knowledge can change during the lifecycle of a product or even during the lifecycle of an order while the configurator takes care of different updating tasks automatically. This thesis shows a way of documenting configuration knowledge efficiently while the configurator built on top of the configuration knowledge handles the changes related to the configuration process and configurable products and furthermore gives a good understanding how producibility is affected by configurability and modularity.

1.1. Research problem

Configurable products offer many companies effective means to meet customer needs more effectively. Configurable products affect the entire organization in many ways. Firstly, product development needs to be designing configurable products usually by the use of modularity. Secondly, the production system needs to be re-engineered to meet the needs of configurable products. Thirdly, marketing needs to understand the modular systems produced by product design in order to sell customer varied products. Finally, after sales needs to maintain the produced product structures in order to serve customers after the product has been delivered. Thus, the entire organization is concerned while the integrative aspect is configuration knowledge related to the configurable products.

As configurable products are considered, large amount of knowledge is related to the individual products. The problem is that this knowledge changes as time goes by, i.e. during the lifecycle of product families the knowledge related to these product families needs to be maintained. Even if companies have configurable products, they can sell and design them, the knowledge related to these product families is not usually well documented and maintained adequately enough. Also the use of configurators is not relevant if configuration knowledge is not maintained systematically. Configuration knowledge of configurable products is very important for companies using strategies based on configurable products. The problem is that while companies are developing their processes and products towards making configurable products possible, there should be an easy and systematic way of understanding, documenting and managing this type of knowledge. Moreover, modularization is heavily related to configurability and producibility while these issues are often misunderstood.

Finally, the configurable products are clearly an issue concerning all the levels of organization. Understanding the modular system used in the company's product is essential since modularity determines configurability of products. Modularity is considered to make configurable products possible, but that is only one use for modularity, i.e. modularity should be seen holistically enabling configurable products next to the similarity between product structure and production system. The problem is to achieve both simultaneously while it is largely determined by the capability of the product design to provide modularity. As modularity needs to be seen holistically, Holmqvist and Persson (2004) also see that modularity is not only a product issue, but processes and product design are heavily involved. Changing the type of modularity changes the processes and organization of product design. As companies face problems with the type of modularity selected, the change process can be problematic and even impossible if the knowledge of the current situation is not understood. The product structures of configurable products needs to be systematically designed and delivered to the organization trying to avoid sub-optimization, i.e. the consequences of modular system to the organization needs to be understood in order to deliver adequate designs.

To sum up, the problem domain can be presented as including the following main issues:

- Documenting configuration knowledge
- Maintaining configuration knowledge
- Establishing modular product architectures in the context of configurable products and producibility
- Making the use of configurators possible in changing environment
- Understanding the impact of configuration knowledge to organization

1.2. Research questions

The main goal for this research is to establish an easy method for presenting configuration knowledge in modular product architectures. Modularity, product structuring, configurability and product development processes form the basis for configurable products whereas the developed method takes care of the lifecycle maintenance issues as well as development of modular structures by understanding the current situation. The goal is to give an understanding of the importance of configuration knowledge and its capability to integrate different functions of a company.

While configuration knowledge affects all the stakeholders in the company it is necessary to address the importance of product structure and its modularity. In many cases modularity offers means for configurable product structures, but this is only one aspect of modularity though it can be seen as one of the most important while configurable products are considered. The idea is to connect configuration knowledge with types of modularity in order to further develop product modularity using configuration knowledge as well as to study the formation of product structures during the product development process. The importance of configuration knowledge considering different operations is addressed and revealed by the development of various software tools based on configuration knowledge. Finally the product structure and modularity are considered in the context of producibility of configurable products.

From the above discussion the research questions have been defined as follows:

- By what means the configuration knowledge can be effectively documented and managed?
- What type of configurator can be used effectively in a changing environment?
- What is the relationship between the configuration knowledge and the organization?
- How is producibility related to configurable products and modularity?

The goal is to maintain and use the defined configuration knowledge over time and give the case company the means to effectively utilize the knowledge related to configurable products. One of these goals is to use a product configurator effectively while simultaneously using configuration knowledge with a broad range of applications.

1.3. Supporting hypotheses

The main problem in establishing the configurator is the management of the configuration rules and maintaining them during the lifecycle of the product family. Even if the configuration rules can be established once, the problem is the tedious maintaining task that is mainly due to the changes occurring in the modules and their interfaces. Furthermore, problems are even more significant when a configurator is used. From this discussion the first hypothesis is derived as follows:

- Using configuration matrices, it is possible to present the generic product structures and the dependencies of the modules and saleable features so that the configurator itself and also the product structure can be maintained during the lifecycle of the product

As the generic product structure is presented and all the configuration rules are defined, the point is to make the tool useable. The second hypothesis is derived:

- It is possible to develop a software system to interpret the matrices in order to automatically handle the updating task related to the configuration knowledge and to provide an effective product configurator

When there are the matrices, generic product structures, configuration rules and the software system to maintain them, the concentration should be focused on the analysis side of the matrices and their structures. The third hypothesis is derived as follows:

- There is a possibility to use the generic product structures and rules of the matrices to derive tools for analyzing the products and to use the configuration matrices as an integrative element for the organization

As the matrices are developed and used as part of the routine work, the benefits start to appear. The knowledge stored in the matrices provides a strong base for the configuration task, but also for the analysis side.

1.4. Research method

Research methods are usually divided into basic and applied research. Järvinen (2001) sees that the aim of basic sciences is to describe the part of reality whereas applied sciences use the results of the basic research while the goal is to achieve a wanted final state. This research is clearly applied research since there are well defined problems as presented above to be solved by using available information.

1.4.1. Constructive research

Constructive research tries to solve a specific problem or to derive a method to solve certain kinds of problems (Olkkonen 1994). For constructive method creativeness, innovativeness and heuristics are central. According to Olkkonen (1994), creativeness and innovativeness are sources for construing the solution method whereas heuristics are used to solve a problem step by step. Järvinen (2001) concludes that "It is typical for constructive research to build a new innovation and this process is based on existing (research) knowledge and/or new technical, organizational etc. advancements." Thus, the main goal is to use existing understanding and reach a final state by making new innovations of the process or by constructing new artifacts or innovations that solve the problem, i.e. the wanted final state can be reached. Olkkonen (1994) adds that the outcome of the research is evaluated usually by a case study which validates the created artifact or solution.

1.4.2. Design science

Olkkonen (1994) sees that constructive research has strong features of applied design science. Design science is defined by Järvinen (2001) as follows:

- "Whereas natural sciences and social sciences try to understand reality, design science attempts to create artifacts that serve human purposes. Design science is technology oriented."

Järvinen (2001) suggest that design science has two activities which are build and evaluate. The building activity includes the construction of an artifact or an innovation while the evaluating activity determines the performance of the artifact or the innovation. After the building phase the artifact is used. Next to the tasks of building and using Järvinen (2001) also considers demolishing, thus the aim is to understand the lifecycle of the artifact or innovation. The task of demolition of an artifact means the end of the lifecycle where it can be either taken away from use or replace the artifact with a new one. The demolition phase should also be used as part of the evaluation criteria. The model that Järvinen (2001) presents is shown in Figure 1. This includes the framework for building and evaluating the innovations.

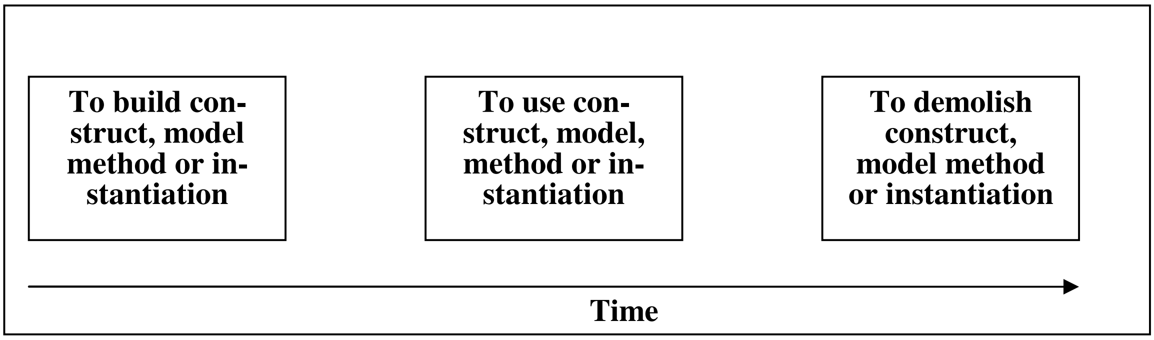


Figure 1. Lifecycle model for an artifact or innovation (Järvinen 2001)

Järvinen (2001) states that the reason for building new innovations is either the fact that this type of innovation is not available or the old innovation produces low quality outputs. The main idea to build an innovation is to take a step from the initial state to the target state as presented by Järvinen (2001) in Figure 2.

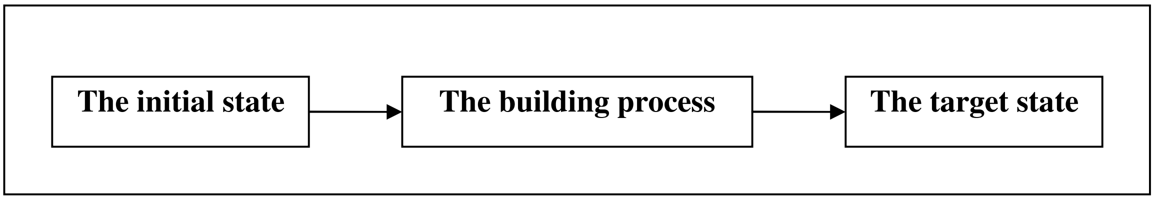


Figure 2. Building process for an innovation (Järvinen 2001)

For the initial state the performance of an artifact can be considered next to the defined goals. There is also a possibility that existing solutions or artifacts can be used in a different way to reach the goal. The initial state needs to be defined in order to have a comparison state for which the reached results for the new innovation can be compared to. For the target state Järvinen (2001) defines the following alternatives:

- The target state is known
  - researchers try to implement the change
- The target state is unknown
  - specify the target state and then implement, or
  - accomplish target-seeking and implementation in parallel

Next to building a new innovation there is also a possibility to buy one from the markets. To conclude the building process Järvinen (2001) provides the following process shown in Figure 3.

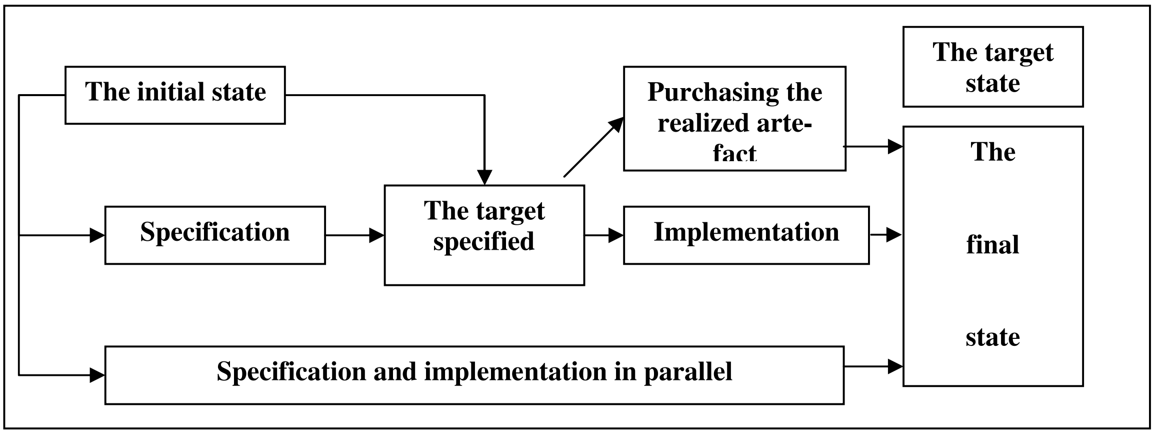


Figure 3. The building process (Järvinen 2001)

The specification process in Figure 3 is meant to produce a model of the target. The implementation process takes care of the building the artifact or innovation. Only purchasing an artifact can be scientifically a small contribution, but it is possible. Finally the parallel process for specification and implementation is usually the case when the researcher cannot imagine the final output. This is usually the case when such an artifact has never been established. The idea is that the people learn by doing and new revisions of the artifact are established every time something new is learned during the research (Järvinen 2001).

For the evaluation of the artifact, Järvinen (2001) provides two possible alternatives. In the first one the artifact is totally new and the researcher can consider its usefulness and benefits or to consider if the construct better defines a phenomenon studied. The next possibility is a situation where construct, model, method or instantiation already exists. In this case the researcher can consider if the new artifact is better than the old ones. According to Järvinen (2001), the evaluation phase is central since it needs to provide the insight if the accomplishment is good enough, i.e. has there been enough improvement due to the new artifact.

1.4.3. Conclusions of the research method

Constructive research is a subset of design science. According to Olkkonen (1994), even if constructive research resembles design task, the feature that makes constructive research as design science is the fact that the purpose is to create new methods to solve design tasks. This research is trying to accomplish a new way of documenting configuration knowledge both for existing products and new products as well as to provide a framework to consider configuration knowledge in the organizatorial context. The framework uses different artifacts and methods including a configurator built during this research. Thus, the above reasoning of constructive research and applied design science holds true in this research. Furthermore, the process defined in Figure 3 describes the process used in this research. First of all, the final state was partially defined and the initial goal for the researcher was to reach that final state. While the research advanced, the final state was modified frequently to provide even better results. Thus, also the specification and implementation was done partially in parallel. This was mainly possible because the researcher was located to the case company during the research. This made heavy iteration possible with the organization to deliver suitable solutions.

1.5. The scope of the research

Configurable products are the main domain for this research while modularity is considered the enabler of these types of products. Modularity is well understood while the relationship between modularity, configurability and producibility is more challenging when considering modularity holistically. The integration between modularity and configurability is considered in the context of producibility in a make-to-order environment.

Even if product design has good capabilities to design modular product architectures, changes take place during the lifecycle of the product and also during the lifecycle of an order. For configurable products this means that the maintenance of the configuration knowledge is central. Thus, a dynamic environment for configuration knowledge and modular system is considered during this research and the requirements for the configurator and configuration knowledge management are based on this environment. This also includes the dynamic nature of the customer needs during the order-delivery process. For both of the cases it can be argued that if the processes are capable enough there are no needs to make changes. While considering configurable products the companies are trying to provide flexibility for customers to select, thus also the changes can take place within predefined limits. The emphasis is to provide robust processes and tools to provide this flexibility systematically.

Considering the configuration knowledge and the organization, the main emphasis is to provide tools for order-delivery process. This process mainly includes production, marketing and sales, and purchasing. As the concentration is on configurable products and their modularity, the emphasis is on the product development and its processes as well. Thus, only after sales is left with less consideration in the context of configuration knowledge.

For order-delivery process a make-to-order environment is considered. For this thesis make-to-order environment is understood to be a customer specific manufacturing environment, where there are also part and component manufacturing next to assembly operations. The configurator built during this research is providing solution for a dynamic environment for make-to-order processes. It is assumed that products that need design tasks during the order-delivery process are considered separately outside the scope of this thesis.

The products considered are heavily based on the case study of this research. Products for the case are complex mechatronic machines that include mechanical structures, software systems, hydraulics and electronics.

Finally producibility is considered in the context of configurability and modularity. Producibility integrates these factors and provides means to offer inputs to the above mentioned stakeholders to understand modularity and configurability more broadly. Producibility is one of the main focuses during this research. In this context producibility is provided by product design through configurability and modularity while the order-delivery process mainly harvests the benefits of producible products. When modularity is considered in the context of configurability and producibility simultaneously the view of the modularity needs to be broad.

1.6. Contribution of the research

The contribution of this research is divided into contributions to the scientific community and industrial applications. The main contribution is the increased understanding of the integrative nature of the configuration knowledge as well as the importance of configuration knowledge maintenance in order to provide means to maintain a configurator in the changing environment.

For the scientific community the contribution can be seen as an insight that this research gives to the importance of configuration knowledge for companies designing and producing configurable products. Issues like configurability, modularity, product design processes and producibility are all considered. The developed method can be seen as an integrative tool between different operations in the company and the aim is to holistically consider the use of configuration knowledge and modularity. Another view of the research is the process-MSDD (Manufacturing System Design Decomposition, see the description of the project in Appendix 1) project which this research is part of. Considering this research contribution, Process-MSDD gives the boundaries in the context of producibility. Producibility is considered to be in the heart of the research as the configuration matrices glue together different departments of the company in the context of the product architecture. The ideas of the modularity and configurability in the context of producible product also give feedback to be used to further develop the models of Process-MSDD project. For scientific community the main contributions are:

- Considering modularity and configurability in the context of producibility
- Providing a method to document configuration knowledge and use it to provide understanding of modularity and configurability
- Providing a framework for configuration knowledge based on established tools
- Providing requirements for a configurator in changing environment

The second place where the contribution can be realized is the industrial applications in general. The main goal is to develop a method easy enough to be used in real applications. One of the main drivers for this research is the case company and its needs, i.e. the method is used simultaneously as it is developed. The benefits of revealing the configuration knowledge should be clear for industry, not only for enabling the use of configurators, but more broadly to understand the concept of producibility of product architectures. For industrial applications the main contributions are:

- A clear and an easy method for documenting configuration knowledge
- Clear understanding of producibility, configurability, modularity, and how these issues interact
- Clear understanding of the importance of configuration knowledge to organization
- Understanding the need for concurrent engineering to provide producible products
- Provide a configurator with minimal updating tasks
- Tools based on configuration knowledge

1.7. Outline of the thesis

The rest of the thesis has been divided into chapters as follows:

- 2. State of the art
- 3. Configuration matrices and its implications to organization and its processes - the developed method
- 4. Case study
- 5. Results
- 6. Discussion and further research
- 7. Summary
- 8. References

The state of the art section provides insight into the research field of interest. There are five main themes running through the state of the art section as follows:

- Modularity
- Product configuration
- Producibility
- Processes for modularization
- Matrix presentations and methods

The themes in chapter 2 are selected to support the framework presented in chapter 3. The main idea is to work in the context of producibility and, as mentioned, considering configurability and modularity. Matrix presentations are studied since matrices are used in chapter 3 to present the configuration knowledge. Finally processes for modularization are considered carefully to select the best one to be used in chapter 3.

Considering the above mentioned themes the developed method is presented in chapter 3. This chapter provides detailed information about the developed configuration matrices, their formation processes and most of all the developed configurator. Configuration matrices are also tied to a best suited modularization process found from the literature.

After the presentation of the configuration matrices and the configurator, the case study is considered in chapter 4. This chapter includes profound discussion of the implications to the case company processes realized by the configuration matrices. Finally, the tools based on configuration knowledge presented in the framework and used by the case company are shortly presented.

Finally chapters 5, 6, 7 and 8 provide the main results, discussion and further research, summary, and references respectively. Figure 4 concludes chapters 2, 3 and 4.

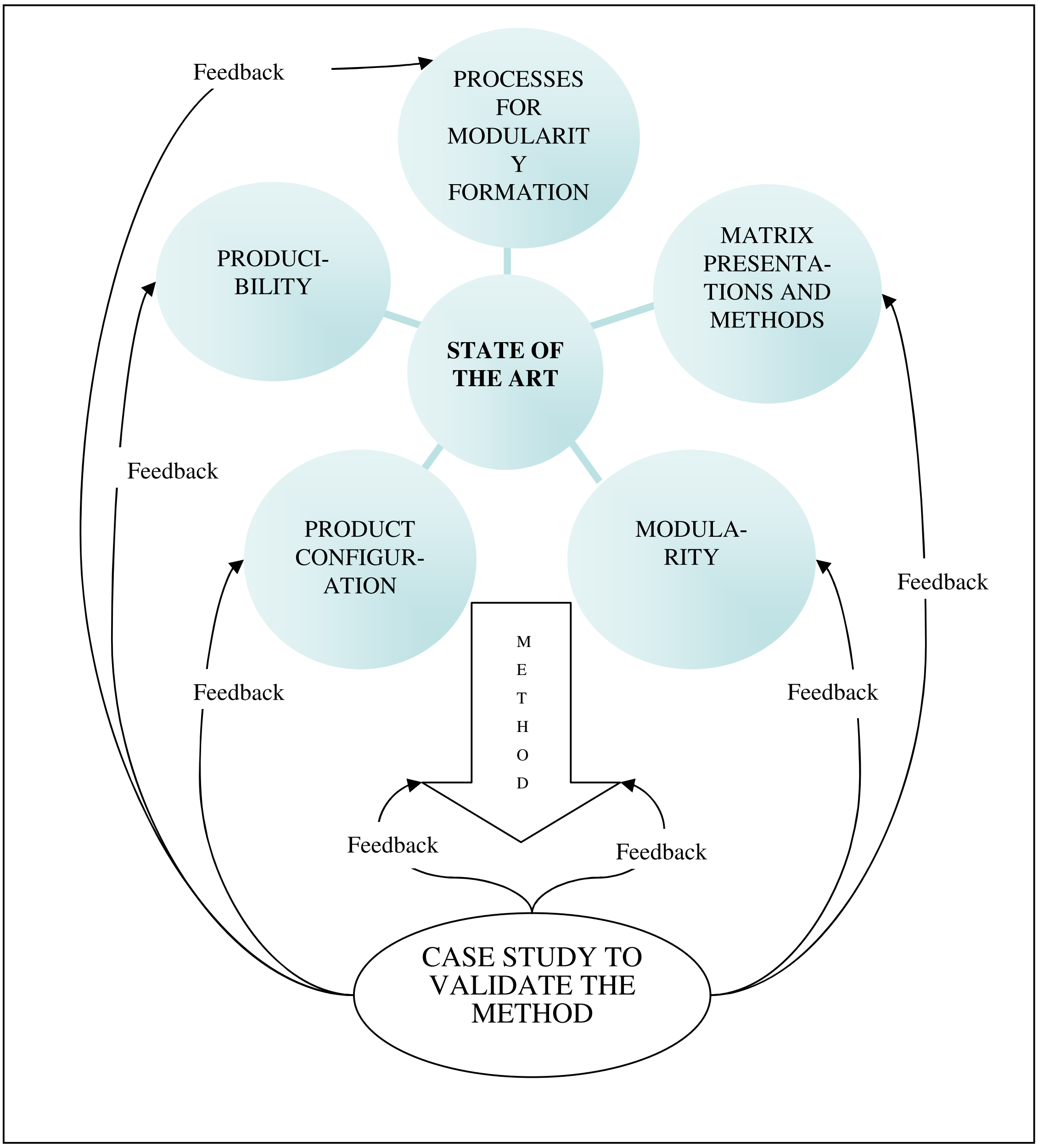


Figure 4. The structure of this research

As shown in Figure 4, the feedback loops are representing the knowledge gathered from the case study and as the method is improved, the processes of the case company are also altered and vice versa. As mentioned previously, the researcher was actively part of the company's operations making the feedback loops and heavy iterations possible. This made the integration between processes and new innovations possible which then provided the results for the dynamic environment.

2. STATE OF THE ART

2.1. Introduction

The main focus of this chapter is to present the field of research that supports this study. The main interest is to derive a good understanding about modularity, configurability, configuration models, product structuring processes and producibility of product families. In the context of this research the questions that should be answered during this chapter are:

- What is the role of configurability when deciding modularity and modules for the product structure?
- At what point should the product structure start to arise during the product development process?
- What types of modules and modularities are found from the literature?
- What types of methods could be useful in determining modules and also configuration models?
- What could be the type(s) of module that could satisfy the needs of the different functions and processes present in a company?
- What types of matrix-based methods are available?
- How is producibility related to product family structures and configurability?

2.2. Product structures and modularity

In this section modularity and the concept of module are discussed. The main focus is to present viewpoints of different researchers on the topic. The idea of modular product structures, types of different kinds of modules, platforms and different types of modularization processes are presented. Appendix 5 presents the effects of modular product structures.

2.2.1. Modular product structures and issues affecting modularity

Modularization is defined by Österholm and Tuokko (2001) to be the decomposition of the product structure into units (modules), which have carefully defined and standard interfaces. This definition also holds the idea of the module, thus a module is a unit with standard and defined interfaces. Lapinleimu (2000) defines modularization to be the actions where product structure is defined by using modules whereas Erixon et al. (1994) defines the same phenomenon to be the act of decomposing the product structure into modules. As all the above definitions use decomposing the product structure into modules, Aarnio (2003) has developed a method in which the final modular structures are derived from the initial modules by integrating them in the context of configurability. MBI (Modularization By Integration) uses configurability as the main driver to form the final modules. Also Erixon (1998) uses integration to form modular product architectures, but uses so-called module drivers to decide the degree of integration of preliminary modules. These and other processes will be discussed in detail in section 2.2.4.

According to Ulrich and Eppinger (2000), a product can consist of structural or functional elements. Functional elements are single operations or transformations that affect the overall performance of the product and physical elements are parts, components and subassemblies that implement the functions (Ulrich and Eppinger 2000). The physical elements are arranged into chunks or structural modules. Hölttä (2004) defines a module as follows: "A module is a structurally independent building block of a larger system with well-defined interfaces." According to Lapinleimu (2000), a module is a structural entity of the final product which has carefully defined interfaces. A module has the following properties (Lapinleimu 2002):

- A module is a changeable component or subunit
- A module is a concrete entity that can be handled as a unit in
  - purchase
  - transportation
  - assembly
  - design
  - marketing

When complex products, and especially configurable products, are considered modular product architectures are imperative in order to handle complexity. Modular products are formed using modules while modules are constructed by grouping components and parts together using the rules and drivers that define the current knowledge and the capabilities of the company concerning the creation of modularity. For this thesis module is following the definitions provided by Lapinleimu (2000), thus module has carefully defined interfaces which provides well defined structural entity in the context of configurability. Structural entity includes the idea of a building block presented by Hölttä (2004). Types of modules in this thesis considers the evolution of modularity (see section 2.2.5), thus the types of modules can be structural or functional while their combinations provide different levels of modularity (structural, functional and customer oriented platform). For example, the product can be divided into functional modularity to support configuration, structural modularity to support production or their combination to provide cost effective variation.

There are many aspects that affect modularity and the types of modules derived from the decomposition or integration phase. Aarnio (2000) considers that even if concrete modules have many reasons that control the integration of components and parts into modules, they are usually related to achieve variety or standardization. These facts are very important and variety can even be imperative to serve the customer efficiently while standardization is related to cost efficiency. Standardization here is related to standardizing the modules shared with different product families while these modules are still used to provide variety to customers. There is also part level standardization which aims to use the same parts and components throughout the product families. What ever the level of standardization is, the production system will be enjoying larger volumes and possibilities to apply various marshalling methods for parts, components and modules when standardization is done properly.

As the concepts of module and modularization are defined, the focus can be set to the modular product family architectures which have the following properties (Ulrich and Eppinger 2000):

- Chunks (module) implement one or a few functional elements
- The interactions between chunks are well defined and are generally fundamental to the primary functions of the product

Pulkkinen and Bongulielmi (2004) argue that product family architecture is "…common structure for each of the configurations…". They also consider the generic product structure (see definition in section 2.3.2) to be a subset of the product architecture. Thus, a product family has an architecture which is realized by the generic product structure. The most modular architecture is the one where a certain functional element is realized by one module and the module has a few well defined interfaces. In this case the design changes can be done without the need of changing the structure of other modules and the modules can be designed independently (Ulrich and Eppinger 2000, Lapinleimu 2000). Ulrich and Eppinger (2000) use chunks to express the definition of module. These chunks are used to establish the required modularity into the product architecture by decomposing the product into the functionalities that the chunks implement.

Modular product structures are usually used when configurable products are considered. According to Tiihonen (1999), products can also be configurable without modular product structure. Configurability is achieved by using parameters in such cases. Considering this research, configurability is achieved via modular product structures. According to Aarnio (2003), the main idea when designing modular product structures should be configurability of the product as mentioned before. If the aspect of configurability is not tackled systematically and understood clearly, there will be problems when defining the customer specific product individuals. According to Lapinleimu (2000), configuration of the product is based on predefined modules that are chosen according to customer requirements during the configuration process. The idea is that no product design is needed when the modules have been designed properly and the customer needs are well defined.

Pahl and Beitz (1986) suggest that modularity is used to rationalize the production and assembly by decomposing the structure into modules, i.e. the modularity is used to rationalize the variability of the product. Also Lapinleimu (2000) considers that the main function of modularity is to satisfy the needs of production and assembly in order to achieve the similarity between the production system and the product structure. According to Lapinleimu (2000), a modular product structure is ideal for production and the similarity between the product structure and the production system is the main goal. For example, outsourcing is not dependent on who makes or assembles the entity anymore when the similarity between production system and the product structure has been established. Exploring the field of modularity the aspect of production is one point of view. Production based modular architectures are very powerful, but there are several factors that will constrain the definition of modules. Whitney (1992) makes the notation that a module can be different from subassembly. Also Baldwin and Clark (1997) consider that companies are trying to expand the concept of modularity all the way to the product design. There is a need to understand modularity more holistically. Marshall and Leaney (1999) define the following features that make the distinction between modules and subassemblies:

- Modules are subsystems that define the product, the production system and the business
- The functional interactions take place rather within than between the modules
- Modules have one or more well defined functions that can be tested in isolation from the system
- Modules are independent and self-contained and may be combined and configured to achieve different variants of the product

Also variety is considered to be one aspect of modularity. According to Marshall and Leaney (1999), a good modular system will enable the use of flexibility designed by the product design in order to achieve customer requirements, production flexibility (module based production system), concurrent production and late configuration of the product. It is clear that modularity of the product structure is not solely about production, handling complexity or to enable variety. Modularity is an idea that concludes the needs of different kinds of departments and processes in a company. Especially because modularity of the product is affecting the entire company it needs to be designed into the product. The problem is when and who develops the foundation for modularity that considers the entire company. Ulrich and Eppinger (2000) also state that modularity is a relative feature. There can be discussion if the structure is more modular than others, but one cannot say whether the product is modular or not.

Aarnio (2003) focuses mainly on the aspect of configurability of the product structure when deciding modules and modularity of the product. Aarnio (2003) argues that the aim of product configuration is to enable variations in products and therefore satisfy the varying needs of the customers. As this thought is considered in the context of the mass customized products, the idea of configurability offers the key driver for the creation of modular products. Mass customization is defined to be a way to provide cost-effective production while simultaneously providing different customer variants (Pine 1993). If modularity is only considered using the production layout as the key driver, problems will occur in the management of the different variants of the product structure. The idea of using the production system as the main principle is one of the far ends of the creation of modular structures. The opposite far end is the idea of using configurability of the structure as the main driver. Aarnio (2003) argues that during the concept phase the company should establish an initial plan for configurability. The information needed for the initial plan for configurability according to Aarnio (2003) is as follows:

- Variety plan
    - from customer requirements
- Commonality plan
    - from standardization and rationalization efforts
- Differentiation & outsourcing plan
    - from supply chain considerations
- Upgrade plan

- from strategic product planning

This planning should be done and established in order to produce good modular structures and wanted modularity. All the ideas and the methods concerning the definition of modular structures seem to start from the idea that the first considerations of modularity will be taken at a very early state of the product development process. Taking into account all the factors that affect modularity, these considerations have to be made at an early stage of the process. In order for the modular structures to work properly, the most significant factor from the above list is the variety plan. Without this plan the customer needs will not be satisfied. The second most significant factor is the outsourcing plan since if the value chain has not been considered and outsourcing still occurs, the modularization needs to be reconsidered and the structure changed. This needs iteration between purchasing, production and product design. The last two factors are not imperative, but considerable and very useful. The above four aspects are all important since they affect modularity of the product. Using the above discussion and the ideas presented about issues affecting modularity, the following figure can be presented for summary (Figure 5).

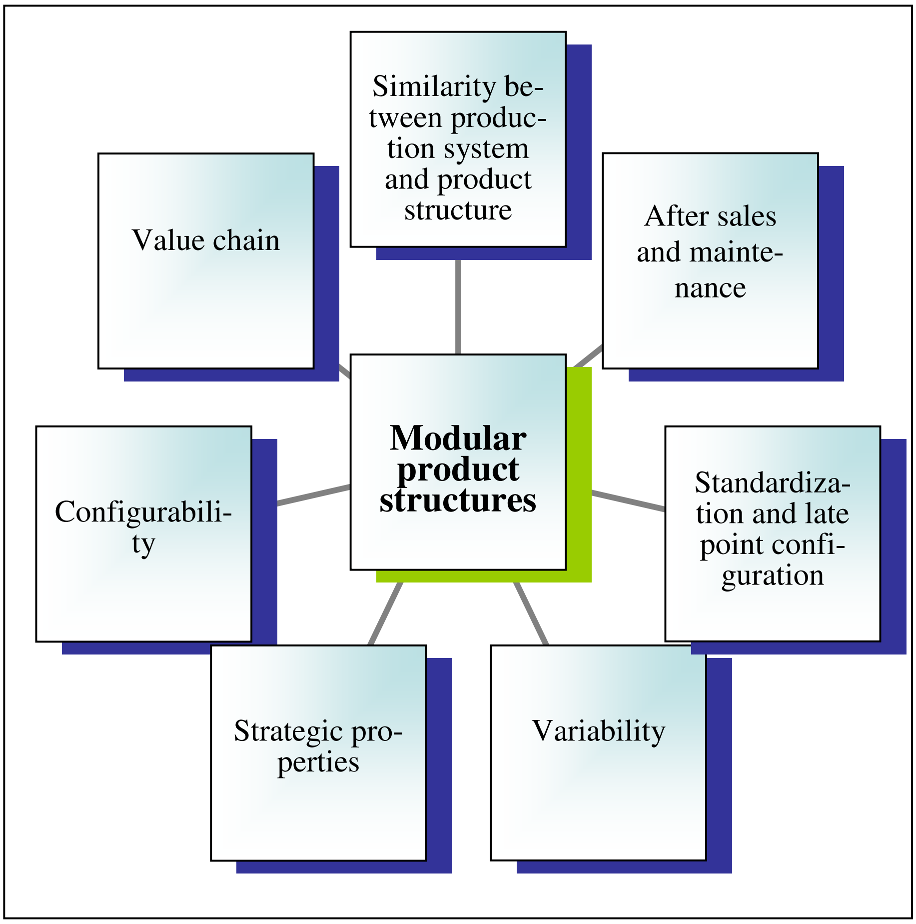


Figure 5. Issues affecting product modularity

According to Figure 5, modular product structures are affected by multiple factors. As stated before, configurability and the similarity between the product structure and the production system affect modularity of the product at the far ends. The impacts of these factors will also have an impact on the types of module derived from the decomposition or integration of the structure. Strategic properties are, for example, upgradeability of the product during its lifecycle (Aarnio 2003).

2.2.2. Types of modularity and modules

Ulrich and Eppinger (2000) divide the types of modularity into slot, bus and sectional modularity. Ulrich and Tung (1991) divide slot modularity into three sub-categories which are component swapping, component sharing and fabricate-to-fit modularity. Table 1 shows the different types of modularity presented by Ulrich and Eppinger (2000) and Ulrich and Tung (1991).

Table 1.Types of modularity

Ulrich & Eppinger 2000	Ulrich & Tung 1991
Slot modularity	Component swapping modularity
	Component sharing modularity
	Fabricate-to-fit modularity
Bus modularity	Bus modularity
Sectional modularity	Sectional modularity

Österholm and Tuokko (2001) define the sub-categories of slot modularity as follows:

- Component swapping modularity
    - at least two different components can be attached to the same base structure
- Component sharing modularity
    - the same component can be used in many different base structures
- Fabricate-to-fit modularity
    - one or more standard components are used with a component that can be parametrically varied

According to Salvador et al. (2002), there is one more type of slot modularity which can be derived from the combination of the above sub-categories. They present the concept of combinatorial modularity. This type of modularity uses standard modules that are connected to each other by standard interfaces. The main difference is the lack of a certain base machine that is evident in other types of slot modularity. The above types of slot modularity are shown in Figure 6.

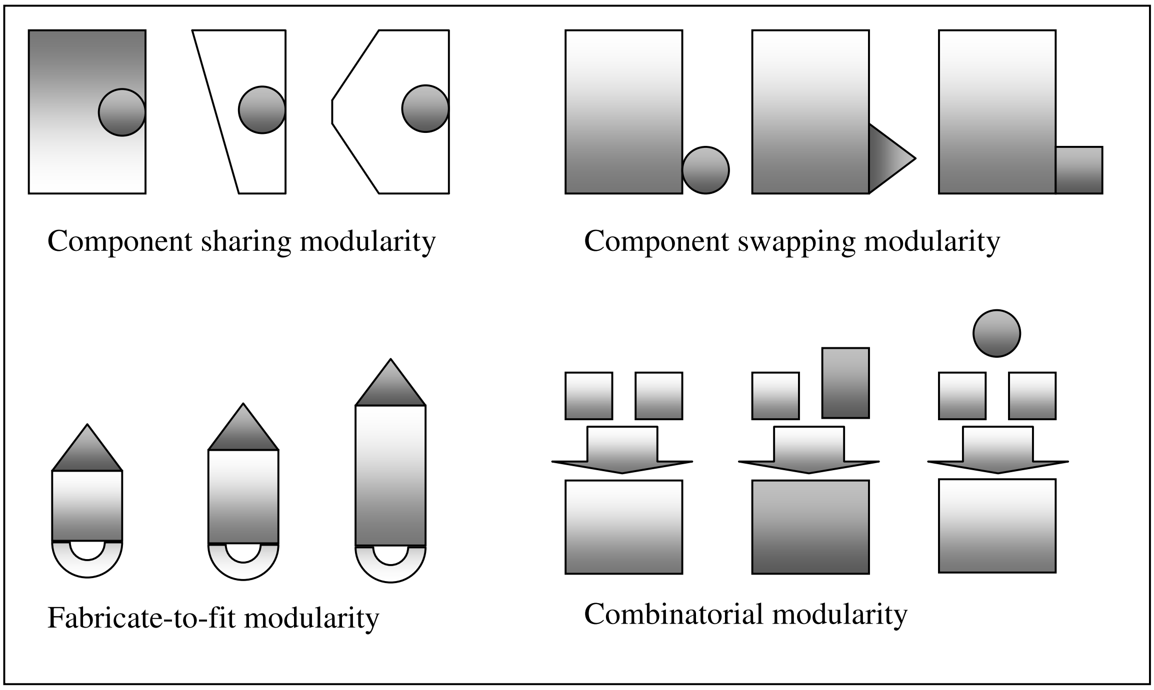


Figure 6. Slot modularity (Ulrich and Eppinger 2000, Ulrich and Tung 1991, Salvador et al. 2002)

In sectional modularity all the interfaces between the modules are identical and the assembly of the product is done by attaching the similar interfaces (Ulrich and Eppinger 2000). Bus modularity has standardized interfaces between the modules and the bus. The modules are attached to the bus which serves as the base for the product (Ulrich and Eppinger 2000). Sectional modularity and bus modularity are shown in Figure 7.

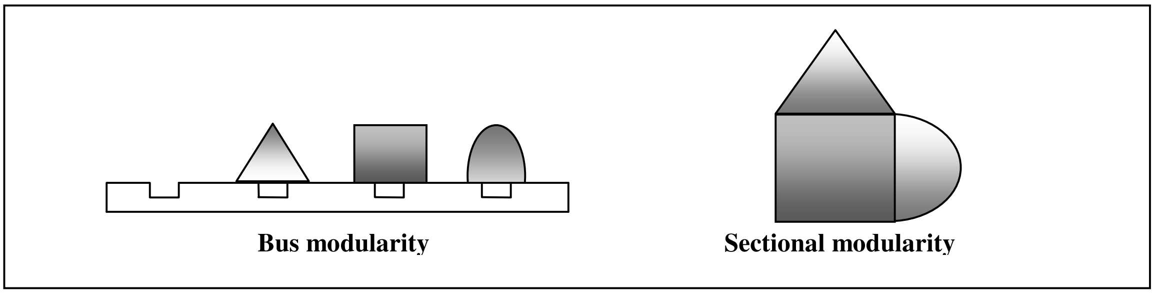


Figure 7. Sectional modularity and bus modularity (Ulrich and Eppinger 2000)

Considering configuration Pulkkinen and Bongulielmi (2004) see that modularity "…enables component swapping in a configuration and component sharing between configurations". Thus, within a configuration the base structure can be varied with different modules to provide different variants whereas commonality can be introduced by sharing these modules between configurations. Note, that all the modules can be predefined and in this sense common, while the different configurations provide the needed variety.

The types of modules are discussed in detail since modularity and the usefulness of the modules are largely dependent on this issue. Pahl and Beitz (1986) divide modules into functional and structural modules. Structural modules are designed according to the manufacturing or assembly, thus they can be assembled by connecting the predefined interfaces during the assembly process. Functional modules are defined according to the realization of the technical functions. The functions are obtained by one module or by a set of modules (Pahl and Beitz 1986). The division to functional and structural modules is due to the drivers that control the decomposition or integration of the product structures, thus the driver is the production system or the functionalities of the product (Stake 1999). Pahl and Beitz (1986) clarify the types of functions and modules according to Figure 8.

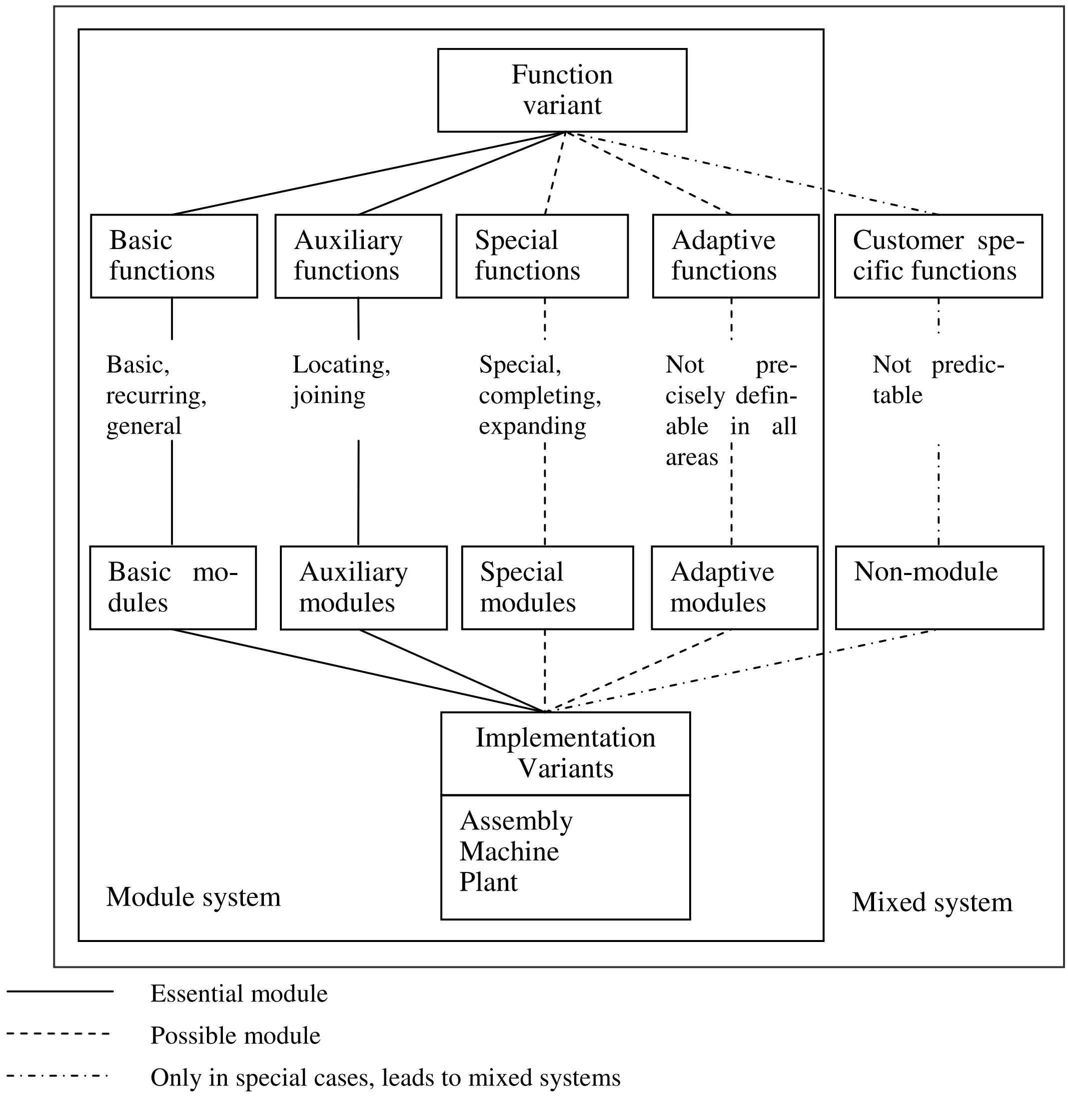


Figure 8. Module types according to the functions (Pahl and Beitz 1986)

According to Lapinleimu (2000), a structural module is a part of the main product and can be handled as an own product itself in production, i.e. a structural module equals the production module defined by Pahl and Beitz (1986). Lapinleimu (2000) divides the structure further to basic modules, main modules and sub-modules. The basic modules are the ones that form the actual product. The concept of main modules and sub-modules is needed if multiple layers in the product structure are required. As Lapinleimu (2000) addresses the similarity between the production system and the product structure, Pahl and Beitz (1986) concentrate on defining the modules via the functions that together form the basis for the product to work. The division of the module types into structural and functional modules represents actually the far ends of modularity as mentioned in the previous section. Both of these types of modularity have their own benefits and shortcomings. This is the reason why the types of modularity should be clarified and understood more broadly in order to define the best possible product structure and modularity for the organization. In this thesis the type of modularity is related to type of module i.e. the structural modules provide the basis for structural modularity while functional modules provided the basis for functional modularity which represent assembly based modularity and function based modularity respectively (see section 2.2.5).

2.2.3. Product platforms in literature

Robertson and Ulrich (1998) see a product platform to be a collection of assets (components, processes, knowledge, people and relationships) that are shared by a set of products, while Meyer and Lehnerd (1997) define a product platform as follows:

- “A product platform is a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced.”

Muffato and Roveda (2000) consider the platform as follows:

- “A product platform is a set of subsystems and interfaces intentionally planned and developed to form a common structure from which a stream of derivative products can be efficiently developed and produced.”

Their definition is an extension from the definition provided by Meyer and Lehnerd (1997). The aspect of intentionally planning the common structures is emphasized. Lehtonen et al. (2004) define the product platform to be a common set of re-usable assets used in developing a set of products to form a product family whereas Juuti et al. (2004) consider the platforms to be treated as technical systems to produce configurations. Juuti et al. (2004) and Juuti and Lehtonen (2004) consider platforms in the context of configurable products to enable efficient creation, manufacturing, delivery and maintenance of variant products.

Stake (1999) defines product platform to be something that is common for a range of products. Stake (1999) does not make a difference of what constitutes the platform and so it can be the parts of the product, the processes that create the product and also the interfaces that define the product. Stake (1999) describes the modular system (includes all elements and all interfaces) to include the product platform (includes elements, interfaces and the common structure) and the variety modules. The product family can be defined from the modular system. A product family includes all the variants that can be derived from the modular system. The concepts of modular system, product family and platform are presented in Figure 9.

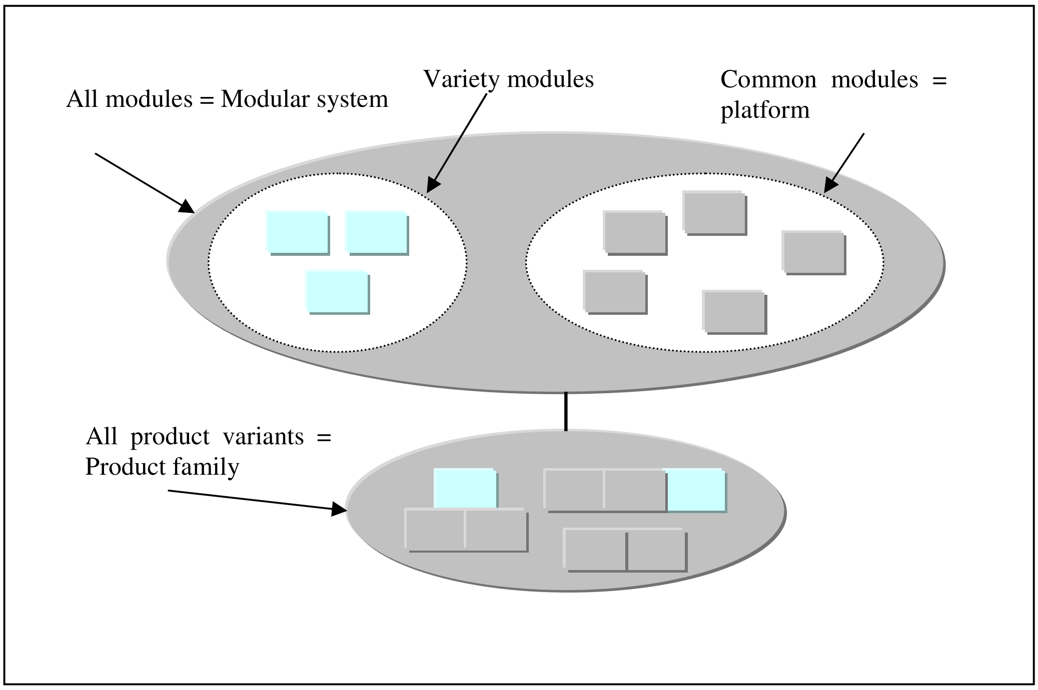


Figure 9. The concepts of modular system, product family and platform (Stake 1999)

As seen in Figure 9, the common modules, i.e. the platform serve as a robust basis for the product family and the variety modules are used to generate different variants for the product family. This thesis uses the guidelines provided by Stake (1999) when considering product platforms. The modular system described in figure 9 is central for this thesis because the generic product structures are based on these modular systems. The idea of providing commonality through platforms while simultaneously offering different product variants for customers is considered in the following paragraphs.

Pulkkinen and Bongulielmi (2004) see that "Product family is a means for a company to manage variety, enhance commonality and reduce complexity in product portfolio and in sales delivery processes". They also agree that the product architecture defines the varying and standard proportion of the product as shown in Figure 9. According to Lapinleimu (2000), the number of variant modules is not critical if:

- All the variants can be produced with the same technology
- The variant modules are known in advance
- All the variant modules have adequate production capabilities
- Production system can reach one piece flow economically

Lancaster (1990) states that product variety is valuable to the individual customer, but at the same time delivering variable products generates costs. This implies that there is an optimal level of variety that is affected by economies of scale and gains from variety. Economies of scale can be improved by using modules in many product families, thus economies of scope helps (Pine 1993). Also Aasland et al. (2001) argue that large amounts of variants makes efficient production difficult and at the same time customers want more and more tailoring into the products to satisfy an individual need. According to Österholm and Tuokko (2001), instead of generating single product variants the platforms should be established in order to offer right kinds of variants to the customers which are economically producible. The platform establishes a basis from where derivative product variants and even new product families can be generated.

Martin and Ishii (2000) see that product variety can be located within the product line being designed and variety across the future generations of the product. They have developed a method (DfV, Design for Variety) for designing product platforms based on QFD (Quality Function Deployment) based matrix presentations and indices that approximate the impacts of variety into the design in order to minimize the possible redesign efforts.

According to Robertson and Ulrich (1998), customers are concerned with what they can get, not so much what is the level of commonality of the product family. They consider the differences between commonality and distinctiveness to reflect on platforms. Muffato (1997) also sees that a platform has many implications as the product needs to be distinctive to the markets while at the same time it needs to be able to be produced at low cost. In order to deliver customer different variants the product family needs to be distinctive and if the costs of the organization is considered, enough commonality needs to be present. According to Robertson and Ulrich (1998), a good product development equals the idea that a family of products is designed and the product developed can be produced in a flexible process tailored to the needs of individual customers. This means that product platforms are designed keeping commonality and distinctiveness of the product in mind. The product architecture can be used to control the level of commonality and distinctiveness to achieve a situation where a high level of commonality can be achieved without much sacrifice in distinctiveness (Robertson and Ulrich 1998). To cope with commonality and distinctiveness of the product, Robertson and Ulrich (1998) present the product plan, the differentiation plan and the commonization plan. According to Robertson and Ulrich (1998), these plans are top-level considerations of product strategy, market positioning and product design, and the purpose is to achieve coherence between them and the three plans.

Meyer and Lehnerd (1997) argue that all product architectures can become platform structures. While all the products have a structure the goal is to make that structure common to many products, i.e. to generate a platform structure shared by the product family (Figure 9). The idea is to create a platform to a product family and then use it as a base structure while designing new product families. Meyer and Lehnerd (1997) see that the product platform creates the basis for the product family, but they also consider the basis for robust platforms. Robust platforms can be divided as follows (Meyer and Lehnerd 1997):

- Insights into the minds and needs of customers and the processes of customer and competitive research that uncover and validate those insights
- Product technologies in components, materials, subsystem interfaces and development tools
- Manufacturing processes and technologies that make it possible for the product to meet competitive requirements for cost, volume and quality
- Organizational capabilities which include infrastructures for distribution and customer support, as well as information systems for control and market feedback

Using the above factors across different product lines creates even more power for the use of product platforms (Meyer and Lehnerd 1997).

Muffato (1999) sees cost reduction, productivity of product development and development lead time reduction as the reasons for platform development. Cost reductions take place in manufacturing and the last two factors in product development. Next to these reasons Muffato (1997) adds benefits such as reduced systemic complexity, better learning across projects and better ability to up-date products. In addition to reasons for platform development presented by Muffato (1997) and Muffato (1999), Muffato and Roveda (2000) consider the following as extra reasons: increased product reliability, increased external variety and reduced internal variety, and increased business flexibility. The last reason presented by Muffato and Roveda (2000) means that platform strategies allow the company to pursue aggressive market strategies due to the reduced times and costs in developing new products. The potential benefits of product platforms presented by Robertson and Ulrich (1998) are as follows:

- Greater ability to tailor the product to the needs of different market segments of customers
- Reduced development cost and time
- Reduced manufacturing cost
- Reduced production investment
- Reduced system complexity
- Lower risk
- Improved service

When considering product platforms Hansen and Mikkola (2004) argue that platforms can be used for many product lines to provide product families through different modular systems. Thus, when product platforms are adequately designed into products, different derivative product lines can use the platform, i.e. the benefits of one platform can be magnified when applying to many products. Juuti et al. (2004) see that the drivers for platforms in configurable products are related to improved sales margin and increase in sales. They see that the former can be reached by efficient use of platforms next to the improved efficiency of delivering product variants to customers while the former is reached by the realization of new customer segments in addition to more variety to current offerings. Juuti et al. (2004) conclude: “The reuse of design, economies of scale and flexible adaptation to meet customer needs are the key elements to platform mode of operations.

2.2.4. Modularization

The formation of modular product structures is discussed by many researchers such as Ulrich and Eppinger (2000), Dahmus et al. (2001), Sekolec et al. (2003), Erixon (1998), Aarnio (2003) and Pahl and Beitz (1986). Their approaches are all considered to provide insights into different modularization methods. Table 2 provides the summary of the methods related to decomposition method used, point of modularization, main driver for modularization and type of modularity. The main impression of these different models is that the creation of modular product structures should be established as early as possible during the product design process. For additional methods for modularization see Jensen and Hildre (2004) and Gershenson et al. (2004).

Ulrich and Eppinger (2000) present a generic product development process which can be used to derive company specific product development processes. As Ulrich and Eppinger (2000) state:

- “The success of the manufacturing companies depends on their ability to identify the needs of the customers and to quickly create products that meet these needs and can be produced at low cost.”

The above statement is a product development dilemma which includes manufacturing, design, purchasing and marketing. Customers are also considering the output and the efficiency of the final product making after sales and the lifecycle costs of the product more important. The work of design is much more than solely producing a good technical design. According to Ulrich and Eppinger (2000), it is the set of activities that is triggered by the insight of the market opportunity and ends in the production, sale and delivery of a final product. The generic process for product development has the following phases (Ulrich and Eppinger 2000):

- Planning
- Concept development
- System-level design
- Detail design
- Testing and refinement
- Production ramp-up

The generic product development process is shown in Figure 10 in detail.

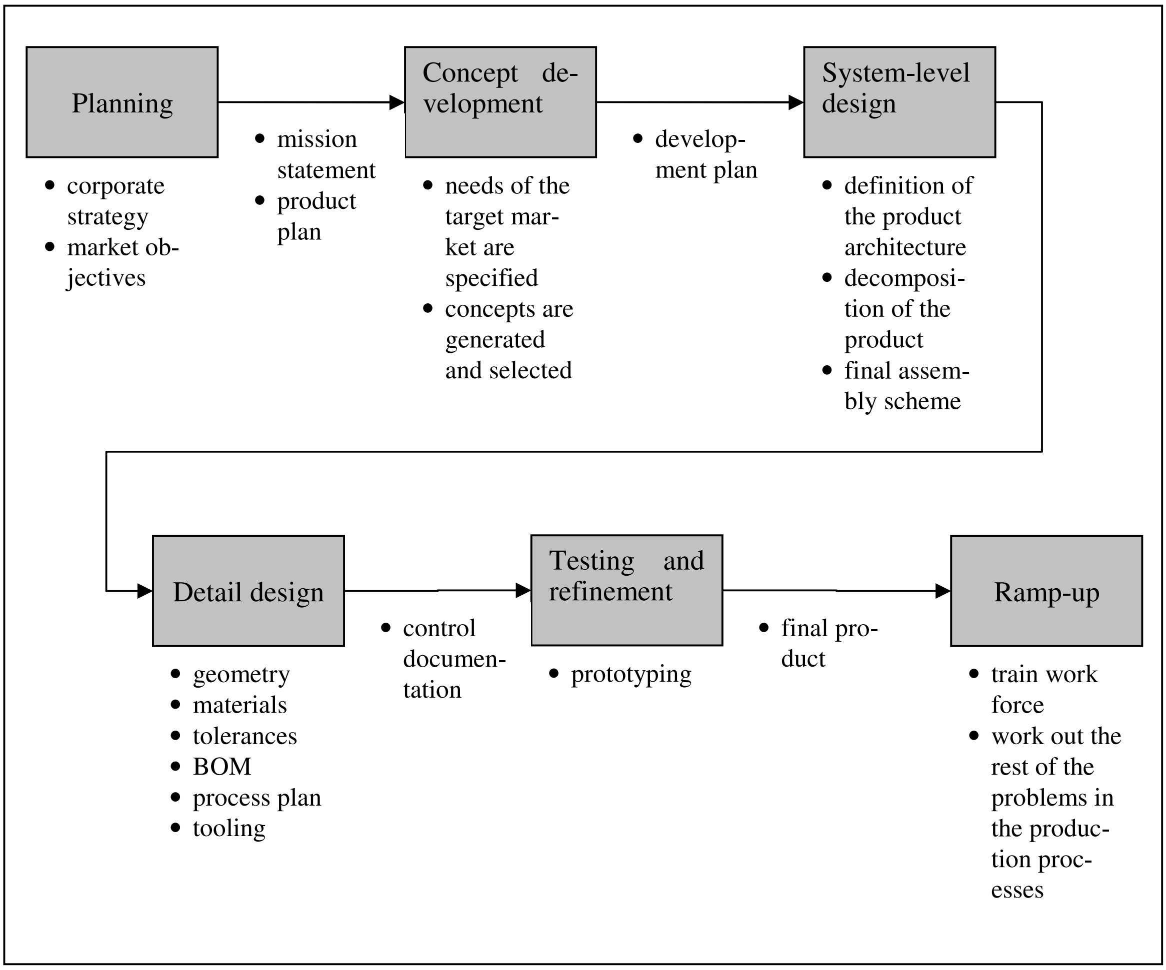


Figure 10. The generic product development process (Ulrich and Eppinger 2000)

As shown in Figure 10, the generic product development process is well defined and the outputs, inputs and process tasks are presented. The main interest in studying the formation of modular structures in this research is to locate the documentation of configuration knowledge into the product development process. This is very closely related to the formation of modules and their variants.

Considering the formation of the product structure in the generic process, the product structure starts to develop during the system-level design phase (Figure 10). At this point Ulrich and Eppinger (2000) suggest that the product structure is defined and decomposed to subsystems and further to components. Even if this is the first point where the product structure starts to form, the planning phase is critical for the rest of the process to succeed. The purpose of the planning process is to make sure that the right drivers are in place to launch a product development project. The output of this process is the mission statement for each of the projects, a statement that covers systematically the needs for the next phases to be successful. The mission statement makes sure that the product development team knows its goals before the project is launched. After the system-level design, the structure is further defined by the design teams responsible for the upper level modules (Ulrich and Eppinger 2000).

During the detail design the development teams come up with all the necessary information needed to manufacture and assemble the product. The output of the detail design phase is the so-called control documentation which includes drawings, tooling, specification of the purchased parts and the process plans. The main thing is to notice the fact that the product structure is ready after these steps. Even the substructures and the variation plans are well established after the detail design. The purpose of the subsequent steps is to make the design better with minor changes to it. The rest of the design process uses the structures and documentation of the detail design phase (Ulrich and Eppinger 2000).

As shown in Figure 10, the testing and refinement process uses control documentation in order to build and test prototypes (Ulrich and Eppinger 2000). Effective prototyping is only possible if the preceding tasks are done according to initial plans and the design has the valid properties.

Considering the MBI method presented by Aarnio (2003), modular product structures are generated at a very early stage of the development process. The aim of this method is to generate many alternative modular structures or modular concepts from where the suitable one is selected. This approach uses configurability of the product as the main driver to establish the modules (Aarnio 2003). Aarnio (2003) presents the idea of initial modules that are derived from the function structure of the product. The final modules are generated by considering configurability, functionality and economics of the integration. The main driver is configurability, but the two other drivers are also considered. The idea is to integrate the initial modules to form final modules and structures in the context of configurability (Aarnio 2003). Configurability of the product structure is actually the driver that limits the size of the generated modules. Aarnio (2003) presents the procedure for the MBI method as follows:

- Establish the initial modular structures
- Analyze the interfaces
- Cluster the initial modules
- Improve the structure

In the first phase the idea is to generate a very distributed structure. The structure at this point is the decomposition of the structure into the functional one (Aarnio 2003). Aarnio (2003) uses the functions/means -tree to accomplish the initial modules. According to him, the function/means -tree is used for the decomposition, but also to create variety to the product structure, thus it is a good tool for creating modularity. The basic idea in the function/means -tree is to decompose the structure so that each function is associated with a solution principle which will realize the function (Aarnio 2003). The means of the function/means -tree represent the initial modules of the structure. These are the modules that will be further analyzed in order to accomplish the final structure (Aarnio 2003).

The second phase is to analyze the interfaces using the integration drivers. The analysis is carried out by analyzing two initial modules and their interface in order to decide if the interface is eliminated (integrated) or sustained (kept detachable) (Aarnio 2003).

The third phase is to cluster the initial modules using the outcome of the interface analysis. According to Aarnio (2003), the integration is accomplished between the modules that have high integration scores, modules that have low score are left detachable and the modules that have medium scores need more consideration. The main driver is the configurability driver. If the configurability driver does not score the highest possible value, reevaluation is needed (Aarnio 2003). The process defined by Aarnio (2003) for evaluating the scores is presented in Figure 11. As shown in Figure 11, the main point is to pass the configurability conformance test (CCT). If the CCT is not passed, the configurability plan is adjusted if possible or the interface kept detachable. No consideration about the other drivers is needed if the CCT is not passed. If the CCT is passed, the other drivers are then analyzed and if the individual scores are high, the integration is possible. If there are low scores the considerations have to be made about the total score of the two lasting drivers. The main point in the analysis phase is that the outcome is a suggestion of the possible modules that can be integrated. Considerations have to be made if the integration can take place or not. During this third phase alternative structures can be produced (Aarnio 2003).

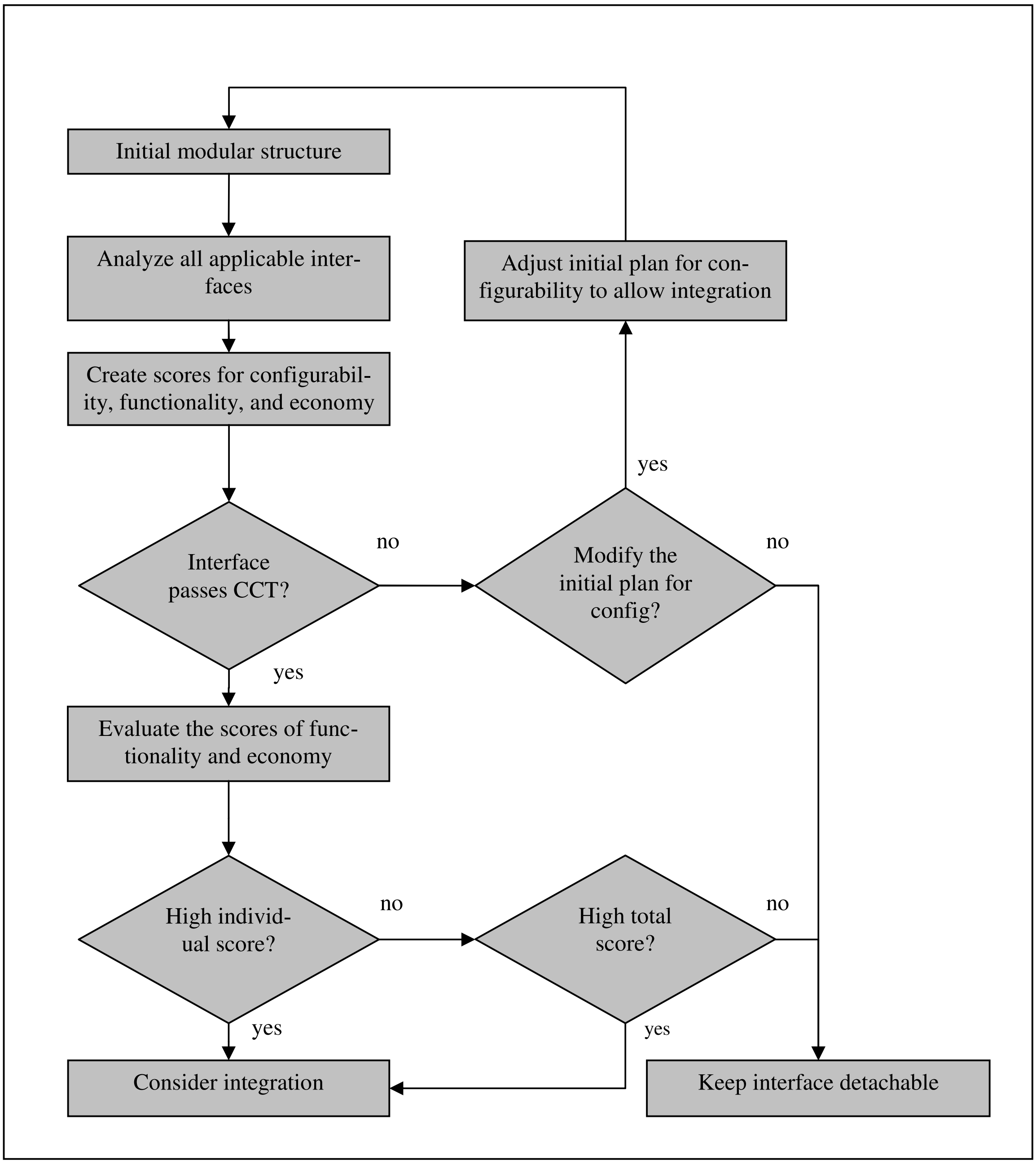


Figure 11. The procedure for evaluating the interface analysis (Aarnio 2003)

In the last phase of the procedure of the MBI method the structures are improved using the data achieved by the subsequent development phases. Aarnio (2003) ties the MBI method on the general product development process presented by Pahl and Beitz (1986). They divide the flow of the product development process into four phases:

- Planning and defining the task
- Conceptual design

- Embodiment design
- Detail design

Aarnio (2003) locates the MBI to start during the conceptual phase after the function structure has been established. Pahl and Beitz (1986) suggest that modular structures are generated during the conceptual and refined during the embodiment phase of the design process. They consider the modularization and the design process to be heavily concerned with the production based aspect. For example, the embodiment design starts with the generation of the assembly structure and proceeds to define the final assembly structure after the preliminary suggestion of the structure has been accepted. Aarnio (2003) suggests that the MBI method is used during the concept phase as mentioned before. Aarnio (2003) uses the general design process presented by VDI 2221 and the ideas of MBI to generate the following design process (Figure 12).

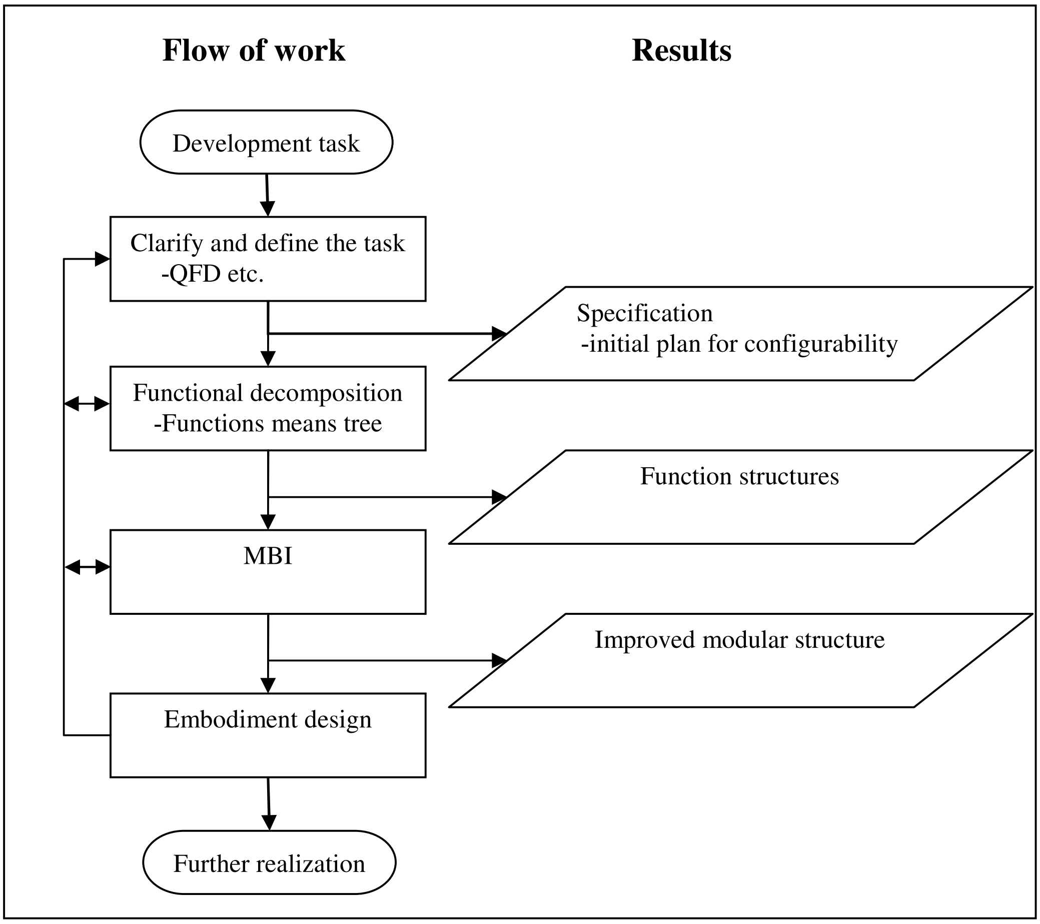


Figure 12. MBI in the context of general design process (Aarnio 2003)

According to Figure 12, the MBI method is used after the functional decomposition is done and the embodiment design starts after the final and improved modular structures have been established.

Dahmus et al. (2001) present an approach to create modular structures and at the same time to offer commonality to the product family. Commonality is accomplished by the shared modules generated during the modularization process. According to Dahmus et al. (2001), an ideal architecture is one that serves the company as effectively as possible, i.e. the decomposition of the product structure should generate useful modules for the company. Company specific reasons make all the modularization processes different from each other and there is no one solution to modularize a product, thus many alternative modular structures need to be accomplished. Dahmus et al. (2001) also state that this process is considered to take place after a certain technology principle is selected and the aim is to concentrate on the deployment of the technology into the product lines. Dahmus et al (2001) present a process for creating modularity and modules during the design phase as follows:

- Develop separate function structures for each product concept
- Union multiple function structures into a single family function structure
- Construct a modularity matrix
- Use the modularity matrix to construct different product and portfolio architectures

The above process is dealing with the decomposition of the product structure and Dahmus et al. (2001) also have research and concept development before the modularization process and a subsequent selection process to determine which product structure generated is best suited for the company.

The process starts with the generation of function structures to every concept generated during the concept phase. The aim is to decompose the product into sub-functions and connect the sub-functions with the various flows possible. The next phase integrates all the function structures generated during the first phase into a family function structure (Dahmus et al. 2001). This is the way that the product family can be observed as a whole and the similar functions across the product line found to form a platform. The modularity matrix is formed by listing the functions from the family function structure into the rows of the matrix. The products from the family are placed into the columns of the matrix (Dahmus et al. 2001). Now the functions can be observed through the entire product family and the values for the functions can be thought by considering the entire product family. According to Dahmus et al. (2001), the matrix can be used to help decomposing the structure into modules for the product itself and also for the product family. Using the modularity matrix, family function structure and modularity rules (presented by Stone et al. 1998), the product family architecture can be established. The modularity rules are used to define possible modules from the function structure. Stone et al. (1998) consider three types of rules which are dominant flow, branching flows and conversion-transmission. All the rules concentrate on analyzing different kinds of flows to determine possible modules. Using the modularity matrix, family function structure and the modularity rules, the product architecture for the portfolio can be determined. While considering the modularity rules and the functions in the matrix, the product modules can be determined and also the product family architecture and shared modules established. Considering the way the modularity is accomplished to the product family, the matrices and the family function structures can become fairly complex when deciding modularity to a large system.

Sekolec et al. (2003) present a process for product structuring using a special matrix approach (the method is discussed in detail in section 2.4.4.) to handle and analyze the internal and external variations of the product in order to generate modular structures. The process has been built around the K- and V-Matrix presentation (Bongulielmi et al. 2001) and used to generate modular structures as early as possible during the product development process. According to Sekolec et al. (2003), external variety is the variety experienced by the customer and the internal variety is the product family variety generated into the company. They also address the importance of generating the modular structures as early as possible during the product design process. The process introduced by Sekolec et al. (2003) is presented in Figure 13.

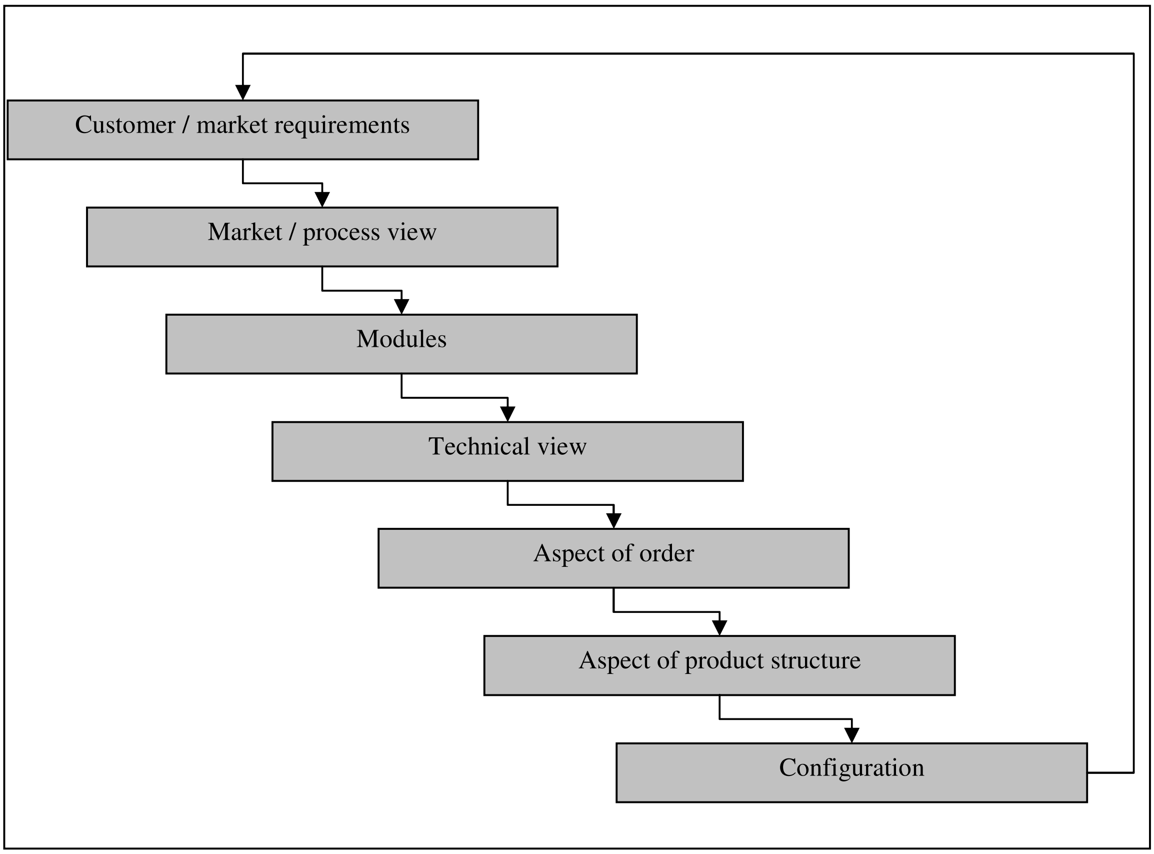


Figure 13. Procedure for product structuring (Sekolec et al. 2003)

The basic idea of the process introduced by Sekolec et al. (2003) is that the customer options or requirements are presented in one matrix and the technical view is presented in another matrix. Considering Figure 13, the modules are generated after the customer needs have been established and the options defined. The basis for the module structure is the functional decomposition of the product. At this point the aspects affecting modularity (production, assembly, customer demands, laws, future developments) need to be taken into account. According to Sekolec et al. (2003), the essential element of the technical view is the product structure and modules. When the matrices are developed both for market/customer and technical view, they can be integrated to form a third matrix that presents the aspect of marketing in relation to the aspect of technical features. The last two phases of the process concentrate on the product structure and configuration by considering the standardization, re-use and configurability of the product structure (Sekolec et. al 2003). According to Sekolec et al. (2003), the creation of the modules can be aided by using the methods such as MFD (Modular function deployment, Erixon (1998)). Sekolec et al. (2003) also discuss the benefits from the matrix system to the development of configuration models (discussed in depth in section 2.4.4.).

Erixon (1998) uses the explosion of variants and time as the main issues that put pressure on product development. He also considers the lifecycle of the product and the design for manufacturing to have its impact on the product development. The purpose of MFD is to satisfy the need for a structured design method for designers to guide them through the development process. The goal of the method is to develop robust modular architectures that serve the company for the entire lifecycle of the product (Erixon 1998). MFD has the following five steps (Erixon 1998):

- Clarify customer requirements
- Select technical solutions
- Generate concepts
- Evaluate concepts
- Improve each module

The first step is used to clarify customer needs by using QFD. The main point is to define customer needs so that the product specifications can be determined. The goal for the first phase is to get an idea of the most important customer needs and to define product specifications according to these needs. This phase should be executed systematically and quickly in order to enable the definition of the project scope and also to enable quick time-to-market (Erixon 1998).

The purpose of the second step is to use functional decomposition in order to generate a more technical view than the created market and customer focus based view. The goal for the functional decomposition is to define functions and sub-functions which fulfill the demand and the technical solutions. The demand corresponds to the customers' needs and the technical solutions are the means that realize the functions in reality (Erixon 1998). According to Erixon (1998), functional decomposition forms the basis for the creation of good modular designs. He adds that a good product design starts with a good specification and a good decomposition. Erixon (1998) also mentions that integration of the functions should not be done at this point. Considering the nature of the decomposition of the product structure, many alternative solutions can be available. Erixon (1998) also makes a notation that the selection of technical solutions can be tested considering production goals since all the solutions survived this far are likely to meet the design requirements.

The goal of the third step is to generate concepts using module drivers and the module indication matrix (MIM). Module drivers are the driving forces behind modularization (Erixon 1998). Erixon (1998) defines the following module drivers for the entire product life cycle:

- Product development and design
  - carry-over
  - technological evolution / technology push
  - planned design changes / product planning
- Variance
  - technical specification
  - styling
- Production
  - common unit
  - process and / or organization re-use
- Quality
  - separate testing of functions
- Purchasing
  - supplier offers black box
- After sales
  - service and maintenance
  - upgrading
  - recycling

The above drivers can be considered to be generic and the list can be added with company specific drivers (Erixon 1998). The whole idea in the third phase is to screen the module drivers against the sub-functions in the module indication matrix and to determine the possible need for integration between the sub-functions into the same module (Erixon 1998). To clarify, the technical solution is a function carrier that is used in MIM, i.e. the functional decomposition breaks the product into functions and corresponding technical solutions (function carrier) to be used when defining modules using MIM. The technical solutions can be modules or they can be integrated to form modules or into other modules. According to Erixon (1998), the drivers presented above can be used to form a basis for systematic evaluation of the sub-functions within a product. The use of MIM is possible when the sub-functions have been translated into technical solutions as mentioned above. The purpose of the MIM matrix is to define which sub-functions can form a module and what sub-functions can possibly be integrated. The idea is to weigh the sub-functions against the module drivers. The highly weighted sub-functions can be candidates for modules and low weighted sub-functions can be integrated with the high weighted ones in order to create larger modules. To group sub-functions together there is a need for suitable module drivers from both sub-functions being integrated (Erixon 1998).

Erixon (1998) also presents an ideal number of modules to equal the square root of the average number of parts in one product variant. This estimation is based on the idea that there is a balance between the time required to assemble modules and the time required to assemble the finished modules to each other in the main flow (Erixon 1998). It is important to notice that Erixon (1998) has a very production oriented way of looking modularity. The main point is to generate modular concepts that serve the company product strategies and also the life cycle of the product itself. The use of module drivers is strongly connected to the product strategies defined at a different process than operative product development. To handle the need for different variants the considerations in the module level give a larger view to the life cycle issues due to the module drivers.

The fourth step is dealing with the evaluation of generated modular concepts. Erixon (1998) sees the interfaces between the modules to be of great importance and he suggests a matrix presentation that shows the interface connections (fixed, moving, media transmitting). The matrix is formed so that two ideal interface principles, "Base part" and "Hamburger" assembly, can be identified. The matrix is used to identify the interfaces that need to be better analyzed and improved. Next to the interface matrix Erixon (1998) has developed a thorough checklist that covers the important metrics that influence good modular design. As the strategic product planning generates the product strategies, the operational product development has very clear goals both for product planning and for the good modular products. The goals can be set and the final results can be then analyzed.

The fifth step is meant to improve each module. As many factors influence good modular designs the MIM matrix is used to point out the main driver behind the module in order to design the module according to the driver influencing the module (Erixon 1998). The whole concept of MFD is targeted on the life cycle issues of the product and the point is to define good modular structures with known constraints. The last part of the modularization process is to document all the data into the module specification sheet. This sheet includes description of the module (drawings), technical solutions (from MIM), interfaces with other modules (from interface matrix) and other factors such as test results (from QFD) (Erixon 1998).

The main interest in this section has been to present different views available on producing modular product architectures. One of the main aspects is the definition of the customer needs and the decomposition of the product functions in order to make the generation of modular architectures possible. There needs to be a very well defined product planning process (Ulrich and Eppinger 2000) in order to secure the strategic aspect of the product development as well as the clear goals for product development teams. Considering the formation of the product structures during the product development processes presented above, the following table can be drawn (Table 2).

Table 2. Modularization and types of modularity

	Ulrich and Eppinger (2000)	Aarnio (2003)	Dahmus et al. (2001)	Erixon (1998)	Pahl & Beitz (1986)	Sekolec et al. (2003)	Aarnio (2003) with VDI 2221
Decomposition of the structure	Functional	Functional	Functional	Functional	Functional	Functional	Functional
Point of modularization and generation of product structures	System level design	Can be applied at various stages	After concept level	Concept level	Embodiment design	Can be applied at various stages	Concept level
Main driver for module creation	Commonality vs. variability, supply chain	Configurability	Modularity rules, shared modules, platforms	Number of drivers, mainly production	Structural vs. functional, company specific	Internal and external variability	Configurability
Type of modularity	Structural	Mixed, company specific	Platform based	Structural	Mixed	Structural	Mixed, company specific

First of all, all the decomposition phases of the processes are based on functional decomposition of the products. Different types of tools can be used, but most often there is a function/means -tree approach used. As the different methods and the point of modularization are considered, the main impression is that the modularization of the products should be done right after the technologies used and the need for variety has been established. The pressure is on the product planning process to deliver plans that are according to the product strategies. As Table 2 shows, the design of modular products requires well defined preplanning in order to consider all the possible stakeholders, i.e. the main drivers for module creation need to be clear. Finally, the type of modularity is considered in Table 2. Type of modularity is related to types of modules discussed in section 2.2.2. As mentioned before, the similarity between the production system and product structure implies the use of structural modularity and good configurability implies the use of functional modularity. The problem of defining the type of modularity lies in the fact that there are many aspects to modularize a product. Thus, the type of modularity can be either structural (assembly based), functional (product development and marketing based) or mixed (having elements from structural and function based modularity). Also the processes and architectures used by companies affect the definition, i.e. there can be different views to product structure or the structure can be designed to incorporate only one aspect. Still the different methods can be considered to include structural or mixed (functional plus structural) modularity. Also the level of decomposition affects the type of modularity, i.e. the decomposition can be taken to very low levels of the product structure which means that the decomposition can decompose the product first to functional modules, then to assemblies, and finally to parts and components. As stated before, the modularity should serve the company as well as possible which is one of the main drivers to use structural modularity whereas functional modularity holds less work for design department and fewer benefits for assembly and operations management.

2.2.5. Closing product structure and modularity

The feasibility experienced by the organization using modular product architectures is affected by the type of modularity used. In industry the types of modules are most likely to be structural (assembly based) or even functional (function based) or the structure is based on product platforms. The evolution of the modular product structures used by the industry is presented by Lehtonen et al. (2003) and the idea is shown in Figure 14. For further information about the effects of modular product architectures see Appendix 5.

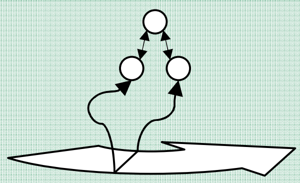
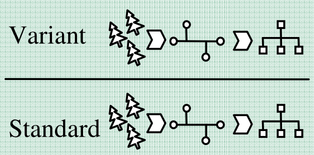
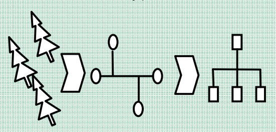
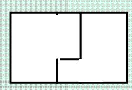
Level of modularity	Implementation	Benefits	Goals
Dynamic modularization that covers the whole life-cycle of product family (Dymo)		Getting grip of the whole lifecycle	Management of change
Customer oriented platform based modularity	Variant / Standard 	Supports company strategy and decreases the cost of customer variation	Cost efficient customer variation by encapsulating the variation in product
Function based modularity	Functions Organs Modules 	Support sales and product development and configuration	Linking customer requirements to the actual modules
Assembly based modularity	Assemblies 	Supports production, procurements and maintenance	Dividing the product according to the production and maintenance

Figure 14. Evolution of modularity (Lehtonen et al. 2003)

Figure 14 is not presenting any relations between the levels of modularity, thus it cannot be stated that assembly based modularity is inferior to function based modularity. The evolution of modularity presented by Lehtonen et al. (2003) conclude the idea of modularity and its connection to the types of modularity. Lapinleimu (2000) and Pahl and Beitz (1986) consider the types of modularity to be either functional (function based) or structural (assembly based). As Lapinleimu (2003) states the types of modules should be structural for the production and care must be taken not to over-modularize the product. Aarnio (2003) suggests that the modularity should be built into the product by considering configurability of the product to be the main driver. While integrating the suggested modules into larger functionalities the primary modular system can be derived. Aarnio (2003) also suggests that all constraints should be considered when creating the modular structure such as manufacturing, maintainability and service.

In the evolution of the modularity (Figure 14) Lehtonen et al. (2003) suggest that the production is best served at the lowest level of the evolution and as the evolution progresses the aim is shifted first to the marketing side to enable mass customization and then to conclude the entire lifecycle of the product. If used right, the platform phase of the evolution possesses the good features of both structural and functional modularity. Considering this phase of modularity the point is that the standard part of the product, the platform modules, can be created to be similar to the idea of structural modules. This point is also mentioned by Lehtonen et al. (2003) and taken even further. According to Lehtonen et al. (2003), the part of the product that is variable needs to be functionally decomposed and the standard part can be assembly based or even integral to achieve the best possible features for the product performance. For Dymo (Dynamic modularization), there is a dynamic platform generated from the business environment's needs. This platform acts as a basis to generate product families and Lehtonen et al. (2004) conclude that the actual product development is done in module creation projects. According to them, the purpose for Dymo is to give cost and time advantage gained by design re-use.

No matter what the main driver for modularity is, or what is the state of the evolution of the level of modularity, or what are the processes for establishing modular product structures, the focus should be how to establish a configurable product structure that has the similarity between the production system and product structure needed. The main issue is to provide enough variety of the product structure for customers while simultaneously offering minimal amount of variety to the production system.

2.3. Aspect of configuration

As configurable products are considered, the processes related to product structuring are important. Product development is responsible for providing modularity and configurability while the documentation of configuration knowledge is important for the subsequent phases e.g. order-delivery process. Pulkkinen and Bongulielmi (2004) see that "Product family design is a process composed of the activities that develop and document product structures and the related configuration knowledge." Thus, more pressure is shifted to product development in order to harvest all the benefits related to configurable products.

McCutcheon et al. (1994) describe a customization-responsive squeeze which means that companies are faced with increasing pressures to offer customized products with short delivery times. Product configuration and configurability are offered as means to deal with complex products in environments requiring customer specific product individuals to satisfy customer needs (e.g. Tiihonen et al. 2003). Careful management of configuration knowledge is an important issue while the organization is also affected (Salvador and Forza 2003, Juuti et al. 2004). Configurability of the product enables the company to shift to a build-to-order environment if all the implications are considered. This section will discuss issues related to product configuration, configuration models, generic product structures and product configurability as aspects of configuration.

2.3.1. Product configuration and configuration models

The configuration process is a procedure in which a customer specific product structure is configured during the order-delivery-process (Tiihonen 1999). Pulkkinen et al. (2004) see that product configuration "…may be regarded as a translation of the specific description of product from one business function to another". According to Männistö et al. (1993), product configuration is a task for creating valid component combinations or descriptions from the specifications. Männistö et al. (1993) consider that product configuration equals configuration task. Product configuration defines a specific product for a specific customer order, i.e. the customer structure based on the customer specifications is established during the configuration process (Tiihonen 1999, Peltonen et al. 1998). Thus, the configuration process uses product configuration to produce the customer specific product individuals. Considering the customer, product configuration gives the freedom of selecting the most appropriate variant of the product from a set of predefined options. The configuration task is defined by Mittal and Frayman (1989) as follows:

- "Given: (A) a fixed, pre-defined set of components, where a component is described by a set of properties, ports for connecting it to other components, constraints at each port that describe the components that can be connected at that port and other structural constraints (B) some description of the desired configuration; and (C) possibly some criteria for making optimal selections."
- "Build: One or more configurations that satisfy all the requirements, where a configuration is a set of components and a description of the connections between the components in the set, or, detect inconsistencies in the requirements."

As Mittal and Frayman (1989) consider the above definition, the most important aspects are that the components are predefined, each component can be connected to other components in a fixed way, and the way of connecting the components is also specified. According to Jorgensen (2001), the task of selecting the appropriate modules is the simplest form of configuration task and in more sophisticated applications the product configuration can be accomplished by selecting values of the properties or by assigning values to the parameters.

Peltonen et al. (1998) describe the concepts and terms of configurable products. This thesis will use the guidelines presented by Peltonen et al. (1998) in describing the field of the product configuration. According to Peltonen et al. (1998), a configurable product is a product family similar to that presented in section 2.2.3. The product family is all the variants that can be established from the platform and the variant modules of the product. The variants of the product, i.e. the product family, are established by selecting adequate predefined components based on the order specification and the configuration model (Peltonen et al. 1998). The order specification represents the requirements of the customer and the configuration model is defined as follows:

- Configuration model describes all the possible variants of the product (Peltonen et al. 1998)
- Configuration model specifies how to create an appropriate variant for a given order specification (Peltonen et al. 1998)

- Configuration model defines a set of pre-designed components, rules on how these can be combined into valid product variants and rules on how to achieve the desired functions for a customer (Tiihonen et al. 1998)
- Configuration model is an abstraction of the real world product family that is specifically meant for configuration purposes (Tiihonen et al. 1998)
- Configuration model (termed product family model) can serve as a foundation for the configuration process because it has a set of open specifications which have to be decided to configure an individual product (Jorgensen 2001)
- Configuration model (termed product model) is a logic structure that formally represents the type of product offered in terms of characteristics (commercial and technical) and constraints between the characteristics (Forza and Salvador 2002b)
- Configuration model (termed product model) sets the rules for dynamically building the product variant documentation starting from the specific needs of the customer (Forza and Salvador 2002b)
- Configuration model is the sum of organized product configuration knowledge that describes a complete product family (Mesihovic and Malmqvist 2004)
- Configuration knowledge is usually captured in configuration models which are used to communicate variation and commonality with pre-defined common and varying structures using elements and relations (Pulkkinen and Bongulielmi 2004)
- Configuration model defines the set of pre-designed components and rules on how these can be combined into valid product individuals (Tiihonen et al. 2003)

Thus, the configuration model is a systematic documentation of the configurable product which makes the configuration process possible. Notice also the definition by Pulkkinen and Bongulielmi (2004) as they already divide the product family structure into standard and varied proportions.

Constraints of the configuration model are used to specify valid configurations from the product family structure (Peltonen et al. 1998). According to Tiihonen et al. (1998), constraints identify the interdependencies between the elements in the configuration model. Without any constraints the configuration model would specify all the possible variants of the structure, i.e. also invalid configurations could be defined. Pulkkinen and Bongulielmi (2004) suggest that usually "…rule or a constraint excludes or includes varying elements so that they have to or cannot exist simultaneously in a configuration." If there is an order specification, the configuration model should include an order specification in addition to explicit structure (see section 2.3.2) and constraints (Peltonen et al. 1998). The validity criteria are defined by the constraints. According to Peltonen et al. (1998), the configuration produced by the configuration process needs to be concrete, complete and valid. They articulate the qualities as the questions asked during the configuration process as follows:

- "A concrete component type asks all necessary questions for a specific configuration, complete configuration answers all these questions, and a valid configuration does not contain any wrong answers."

A concrete component type produces valid and complete configurations if the constraints are valid.

As mentioned before, the specific configuration is based on the configuration model and the model controls the configuration process (Peltonen et al. 1998). Mesihovic and Malmqvist (2004) consider the configuration knowledge to include the marketing product configuration model as well as the technical product configuration model. They see that the marketing product configuration model is used to handle the sales-delivery and after sales lifecycles whereas the technical product configuration model is used to handle the development of new and change of existing products during product development. The configuration model acts as a key factor for the configuration process to work properly. Steger-Jensen and Svensson (2004) see that information management increases heavily when configurable products are considered. Pulkkinen et al. (2004) concluded that while maintaining configuration knowledge is important, it was advantageous to use simple modeling methods to document configuration knowledge. They see that configurability of a product should be considered by "…the minimal amount of exclusive and inclusive relations that can be modeled with simple methods". They also mentioned that sales properties were mapped directly to module level of the product structure to establish the configurations. The purpose of this research is to establish a well defined presentation of configuration knowledge that can be maintained during the lifecycle of the product as well as to study the effects of well established configuration knowledge to the organization. The definition of configuration model is given as follows:

- Configuration model contains all the necessary information (e.g. rules, constraints, modules, options, generic structure) to visualize configuration knowledge for a product family

Obviously the contents of configuration models can vary between different companies as the company sizes and the products differ from each other. The companies also have many different needs while considering the configuration task such as needs to include engineering as part of the sales delivery process. This work is concentrated on a build-to-order environment and the configurations of the product are built from predefined modules which are linked to customer specifications as mentioned in chapter 1.

In order to create customer specific configurations Männistö et al. (1995) define a dream configuration environment to be one that provides a way to effectively model configurable products, support the configuration process and cope with the evolution with the elements needed during the configuration process. This dream environment holds the idea that configurable products need a way to be defined and maintained in a systematic way. The point in this discussion is that the organization needs a clearly articulated configuration model in order to support the configuration process. Tiihonen et al. (1998) make it the responsibility of the product design to develop a clearly understood configuration model during the design process for configurable products. This means that the dependencies of the product structure are defined by the product development. This makes the management and understanding of the dependencies possible which then makes the configuration process possible, even if there is disturbance from the organization affecting the configuration process.

The complexity of the configuration problem is related to the customization of the products needed (Männistö et al. 1993). The configuration of customized products, i.e. products that need product development effort during the order-delivery process is more time consuming and complex than for the products with predefined modules and components. If the configuration process is not isolated from the product design, iteration during the configuration process is possible (Männistö et al. 1993). This is most likely to be time consuming as iteration usually is, and only partially configured products are possible. This is the reason why Tiihonen (1999) suggests that the configuration process and the product development should be isolated. If product structure holds only predefined elements, the configuration process simplifies to the problem of modeling the products and their options and to maintain this knowledge (Männistö et al. 1993). Tiihonen (1999) divides the configuration knowledge into specification knowledge, product individual knowledge and configuration model knowledge. The specification knowledge represents the customer specifications to the product to be configured. Product individual knowledge is used to describe the configured product individual, and the configuration model knowledge represents the configuration models. The division of knowledge needed presented by Tiihonen et al. (1999) can be divided further into the knowledge used by configurator (specification and configuration model knowledge) and the knowledge produced by the configurator (product individual knowledge). The purpose of this thesis is to develop a model that is used to support the needs of the configurator and also to consider the impact of the well-defined configuration models to the different areas of the company.

Männistö et al. (1993) discuss automatic configurators that use predefined configuration knowledge to be problematic when the knowledge of the configuration model changes. This is the problem where most of the configurator implementations fail or slow down. As Forza and Salvador (2002a) (also Peltonen et al. 1998) have found the complex tasks of defining the company's product knowledge in terms of rules and formulae can have a major effect on configurator implementation. They also point out that the biggest challenge after the configurator implementation is the documentation of the configuration models. This implies that if the configurator is set up and the rules and dependencies documented only into the configurator, the problems will occur if changes happen to the existing product families. The problem is that the dependencies are not clearly visible if only documented as rules into the configurator since the modules can have dependencies as well as options selected by the customers. Männistö et al. (1993) see that the problems related to managing configuration knowledge are critical considering the configuration system since if changes take place the system will be outdated as time passes. The maintainability of the configurators is not sufficient enough without a third party information management system. While studying platforms for configurable products Juuti el al. (2004) have found that the knowledge related issues are most important when considering configurable products. Their conclusion is that the company's ability to create, share and utilize configuration knowledge is a success factor. They also consider the maintenance of configuration knowledge to be beyond world class level of competence.

For the product configuration to be efficient there are some aspects to consider. According to Salvador and Forza (2004), efficient product configuration needs a design of configurable product, organizational redesign and support system redesign. The design of configurable products is clear. Organizational redesign is related to getting valid and complete specifications from the sales department. This should be the responsibility of the sales department. Other factors relating to organizational redesign is related to the back office tasks concerning configurable products and the problems of understanding the dependencies between the customer requirements and technical attributes (Salvador and Forza 2004). Support system redesign is needed to systematically present the configuration knowledge in order to manage it (Salvador and Forza 2004). This view is also supported by Mesihovic and Malmqvist (2004). Juuti et al. (2004) consider that capturing the rules and constraints into a configuration model is tedious itself while the challenge is to maintain this knowledge over time. They identify the maintenance of configuration knowledge as an important issue while it is affected by the notation used for documentation next to the complexity of the product.

2.3.2. Configuration models and generic product family structures

According to Männistö et al. (1996), the configuration process uses generic product structures and customer requirements. They also see that the configuration model equals the generic product structure (also Peltonen et al. 1998). This means that the generic product structure describes all the aspects needed during the configuration process. Aldanondo et al. (2000) define the generic bill of materials to equal the physical decomposition model of the product including all the components related to the product possible to be included into the customer configured product individual. According to Männistö et al. (1996), the configuration model is defined by an explicit product structure and by the constraints that controls the validity of the produced configurations. According to Peltonen et al. (1998), the explicit product structure differs from the non-configurable products by the additional optional, alternative parts and parametric components. They also consider that the explicit product structure is based on a hierarchical component breakdown structure which means that the product has components that have their own components as parts, thus the explicit structure can be constructed from components which act as parts in the product structure. For modular product structures the configurations are established by selecting appropriate modules from the generic modular structure according to customer specification. The need to configure in lower levels is eliminated. A generic product structure and its usability is one of the main ideas of this thesis. A generic product structure for modular products is defined as follows:

- Generic product structure includes all the possible modules from the modular system that can be included into a specific customer configured product individual which is a member of a product family

According to the above definition, a generic product structure defines the modular system presented in Figures 9 and 15.

In section 2.2 product platforms, architectures and generic product structures were considered. These issues next to configuration models and modular systems of the product families can be clarified as presented in Figure 15 (derived from Stake 1999, Mikkola 2003 and Hansen and Mikkola 2004).

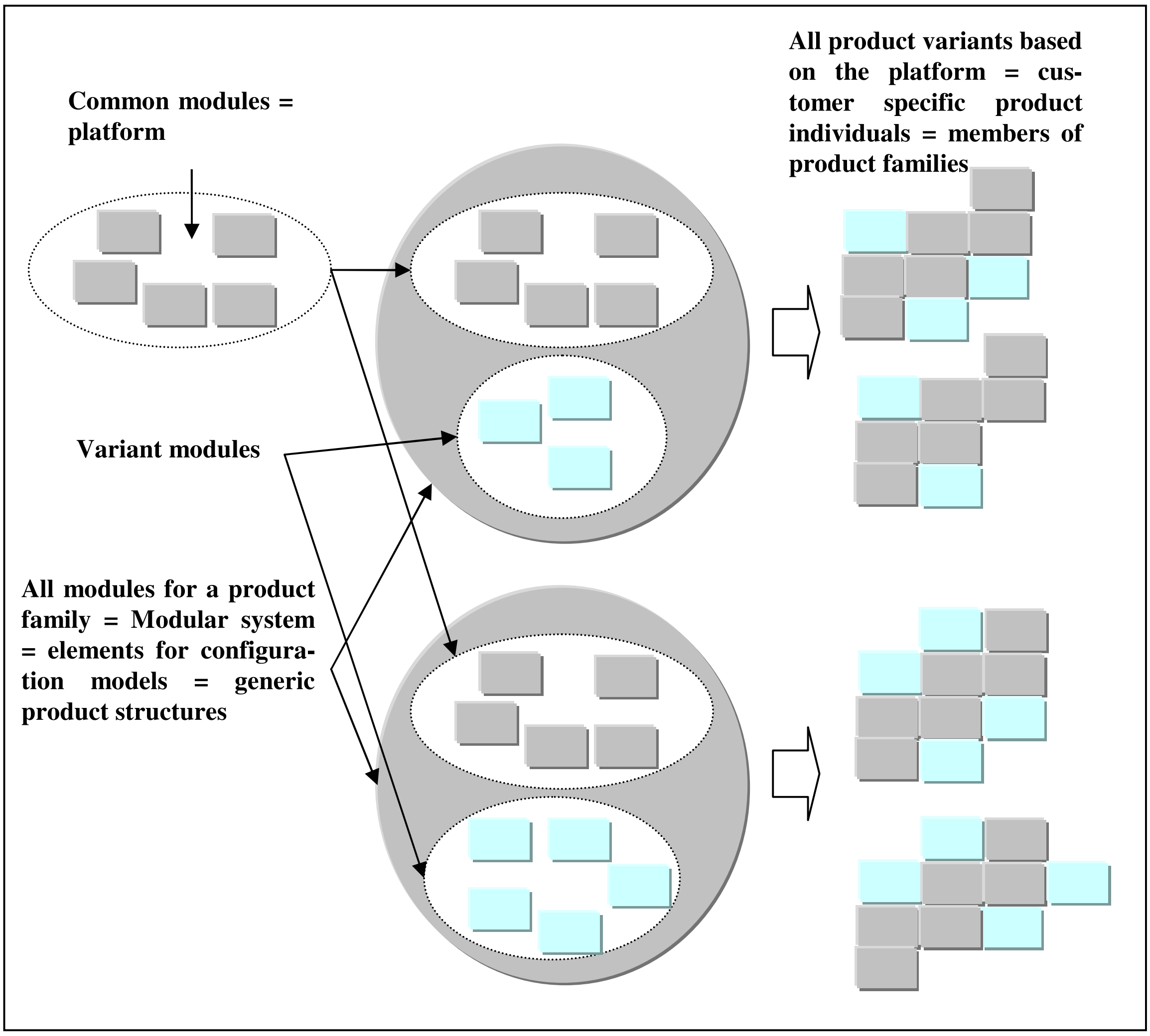


Figure 15. Platform, product family, architecture, generic product structures and configurations (Stake 1999, Mikkola 2003 and Hansen and Mikkola 2004)

Figure 15 shows that the platform can be used to provide a basis for derivative product families. Platform is more than a collection of modules, e.g. issues related to product development processes affect the platform also. Considering the scope of this thesis the platform is considered as basis for configuration models representing a collection of modules. In Figure 15 the product family is considered to equal all the products derived from the modular systems based on the platform. Finally, all the product variants in Figure 15 equal all the configurations possible from the modular system. According to Pulkkinen et al. (2004), "…a configuration is a description of a product individual, which is a member of product family." Thus, a configuration describes a customer specific product individual that is derived from the product family, i.e. from the modular system that the configuration model is based on. There is also a possibility that the platform is only valid for one product family, thus the platform is related to the configured product individuals, i.e. produced product variants that share a common structure.

A generic product structure is used as a basis for the configuration models. When the configuration knowledge is related to the generic product structure, the concept of configuration model is defined. In this sense the generic product structure is a part of the configuration knowledge and when rules and constraints are added, the configuration models are established. The generic product structure also includes the platform part of the product structure. The platform part of a configuration model is defined as:

- The set of modules that will be included into a customer specific product structure every time the product is configured

This part of the product structure is not required for the configuration task, but in order to meet the requirements of the generic product structure the platform part of a configuration model needs to be included. When this platform is shared with different product families, Figure 15 is valid. Next to the capabilities of product design this part largely defines the producibility of the product architecture. Considering Figure 15, the strategic properties of the modular system are considered in the context of platforms. Operational issues surface when Figure 15 is considered further. All the modules in the configured product individuals include the information related to manufacturing, thus the strategic properties of commonality and configurability can be seen to support the emphasis of the production system and the product structure to support producibility.

2.3.3. Product architecture, configurability and sales-delivery process

According to Tiihonen et al. (1998), configurable products have the following features:

- Each delivered product individual is tailored to the individual needs of an individual customer
- The product has been pre-designed to meet a given range of different customer requirements
- Each product individual is specified as a combination of pre-designed components or modules
- The product has a pre-designed general structure
- The sales-delivery process requires only systematic variant design

In a make-to-order environment a company has predefined assets in order to respond to customer needs as soon as possible (Suri 1998). This means that the customer specific product individual will be defined by components or modules that are designed so that the customer variant can be configured from the generic structure as presented by Tiihonen et al. (1998). The ideal situation is when a machine is sold all the parts and components would have lead times so short that they would fit the promised lead time window (Torvinen 2003). This means that the company would only manufacture and purchase the needed parts for the assembly (Suri 1998). This implies that next to tailoring the products according to customer needs there are also implications to the rest of the organization by the product structure. The main point in the above list of configurable product features next to the aspect of pre-designing the structures is that the sales-delivery process requires only systematic variant design, i.e. the product development has been isolated from the order-delivery process and only predefined components and modules can be used (Tiihonen 1998, Salvador and Forza 2002a, Lapinleimu 2000, Jorgensen 2001, Steger-Jensen and Svensson 2004, Salvador and Forza 2004, Pulkkinen et al. 2004).

According to Lapinleimu (2000), the product needs to be configurable because during the sales process the suitable module variants need to be selected in order to specify correct customer specific product individual. This selection process needs the configuration knowledge to work properly. As Salvador and Forza (2002a) have found the key benefits of the product configuration can be diminished if the coordination of the functions is not adequate. For example, the sales personnel need heavy support from the technical side, the technical side is forced to use all their time in checking the validity of the configurations, and false product structures still pass to the production and even to the customer.

Modularity is often considered as the basis for configurability (Tiihonen 1999, Lapinleimu 2000, Pine 1993, Jorgensen 2001). This is the case when the module level of the product architecture is the level where the configuration tasks takes place. Thus, no single components are configured below the module level. Considering modularity as basis for configurability is due to the fact that modular product structures enable the creation of variable functionalities into the product. Modularity provides means to handle complexity of the product structure, thus combining components into larger groups considering the configurability of the created structure. Pine (1993) also considers that the best and the hardest method for achieving mass customization is through the modularization. This means that the product structure is somehow sliced into bunches of components through a company-specific way and the customer specific structure is then configured from these pre-defined modules (Pine 1993). In this way the economies of scale are gained through the components if modularized accordingly. The economies of scope are gained using these modules over and over again in different products (Pine 1993). Pine (1993) describes that modularization is standardization of components in order to increase variety and customization while costs decrease in manufacturing. While these modules are predefined the product structures can easily be configured by combining the existing modules. The problem is that even if the modules are created considering the configurability the customer specifications determine the use of certain modules in product structures. Often the customer specifications have dependencies between each other which make the systematic presentation of configuration knowledge even more important. This actually means that many of the modules are selected by a combination of customer-selected options.

As modularity is often used as a basis for configurable products (Lapinleimu 2000, Tiihonen 1999) other views are also available. Salvador and Forza (2004) consider that product modularity is a subset of product configuration, thus the product structure needs not to be modular in order to be configurable. Salvador and Forza (2004) consider configurability differently from Lapinleimu (2000) and Tiihonen (1999). They see that product configuration only provides different variants, while Lapinleimu (2000) and Tiihonen (1999) see that configurability is a property related to modularity. Thus, the usage of the term configurability is different. Also the definition of the explicit structure given by Peltonen et al. (1998) does not consider modularity at all, but defines the structure to be a refinement of a traditional bill-of-materials (BOMs). Also Tiihonen (1999) argues that modularity is not imperative for product configuration, and the configurability of the product can be accomplished by using parameters (also the definition of extended BOMs by Peltonen et al. 1998). Salvador and Forza (2004) consider that the downside of modularity is the functional mapping into product components as it is very stringent. According to them, product configurability does not need any type of functional mapping, but needs only a relationship between the component and customer requirement in order to define customer specific structures, thus leaving a very complex set of rules and maintenance to the different operations of the company.

According to Tiihonen (1999), modularity of the product can be based on production view, marketing view or on the combination of these two in the context of configurability. This is also analogous to the division of the product structure into structural and functional modules and their possible combination. Tiihonen (1999) suggests that the two views should be kept separate, but still ensure that the connection between the two is secured. According to Tiihonen (1999), the production view of the modularity is accomplished by dividing the product structure to be strictly according to the production. As noticed in Lehtonen et al. (2003) the structural modularity, i.e. the production view is the lowest level of modularity found. This division has many implications, but considering the marketing side the dependencies between the customer requirements and the actual product modules grow substantially and the configuration of the product individual becomes difficult. According to Tiihonen (1999), the problem is that structural modularity is too specific for marketing purposes, the modules have difficult dependencies, and the connection between customer requirements and the modules are not evident. All this affects negatively to product configuration.

Tiihonen (1999) considers that the functional division of the product structure, i.e. the marketing view is optimal for configuration since marketing sells functionalities rather than assemblies (also Riitahuhta 2000, Mittal and Frayman 1989). Using production based division the functions are defined indirectly and the dependencies of the configuration model become tedious to handle. If the product structure is not based on modularity, the product is most likely to be easier to handle or the configuration task is handled via parameters, and the amount of components allows the use of such structures. Considering the domain of this thesis and the type of product that includes thousands of components and even hundreds of modules the use of modularity is unavoidable. If the complexity is not divided into modules, the task of configuring such structures would become impossible.

The modules should be designed so that they have a minimal amount of dependencies between each other to ease product configuration (Tiihonen 1999). The idea is that one customer selection leads to selection of one module or a combination of few modules. The division of modules is very important in the sense that the customer requirements also have dependencies between each other and so the combination of the selected product modules is also affected by the earlier selections and might also be affected by the subsequent selections. This is the field of configuration models to handle. The problem is that this knowledge is usually tacit (Salvador and Forza 2002a) and revealing this knowledge and creating the configuration models can be a very tedious job as the products get more complex. According to Nummela (2003) and Bongulielmi (2002), modules can be tied to the saleable options and by representing these dependencies the rules can be formulated. Modeling the product structure and establishing the dependencies form the basis for successful configuration models and their maintenance.

In the case of functional modularity, i.e. marketing view, the problems occur in production. The problem is that the similarity between the production system and the product is not valid (Lapinleimu 2000, Tiihonen 1999). If the modular structure of the product is not according to the production (assembly based modularity), the similarity between the production system and the product structure needs to be accomplished by other means. Considering the whole concept of product configuration, the aspect of production and product structures should also be answered in order to create producible products.

2.3.4. Impact of product configuration to the organization

According to Jorgensen (2001), the use of product configuration has effects on most of the departments in the company (production, sales and product design). The problem is to draw limits on what to offer and what not to offer to the customer. This is also connected to the ideas of variety in the context of the variety considered by the customer or by the production system (Lancaster 1990, Forza and Salvador 2002b). Product development needs to develop modular products and define the needed information to the rest of the organization including the options to be sold, the release dates of the new possible configurations and the configuration knowledge. According to Jorgensen (2001), the product should be designed to possess a generic product structure to consider all the possible product configurations at an early stage. This also puts the pressure on product planning to deliver adequate information used during the product development. According to Jorgensen (2001), the main impact on sales is the implementation and use of product configuration and on the production the need for minimizing the manufacturing and assembly costs.

Forza and Salvador (2002b) consider the effects of product configuration on the organization to take place in design and engineering, production and sales performance. The benefits are related to the handling of the increasing amount of knowledge associated with the configurable products, elimination of non-value added work and configuration errors, and lead time shortening. Also Tiihonen and Soininen (1997) consider that all the major functions are affected and the shift to configurable products is not only an information system project, thus heavy re-engineering of processes maybe needed.

Mesihovic and Malmqvist (2004) have studied the need for a well-defined process for managing the configuration models. They claim that the main obstacles of using configurable products can be diminished by using a well-defined process that takes care of the needed changes during the entire lifecycle of the product. After the product configuration knowledge has been established and the product has been launched into the production, the configuration model starts to live its own life by reflecting to the changes in the surrounding environment (Mesihovic and Malmqvist 2004). As many aspects affect the configuration knowledge, Mesihovic and Malmqvist (2004) see that companies have centralized the management of configuration models and have teams responsible for the maintenance of this knowledge to enable all the benefits possible from product configuration. The use of effective and well-defined change management is essential for configuration model and process to work.

While economies of scale dominate the world of mass production, the variety dominates the world of mass customization. There is a balance between the economies of scale and the gains possible from variety (Lancaster 1990). The problems from increasing the level of variety are such as the reduction in the volume of production lots, the increase of the design workload connected to the development of numerous new product variants, the proliferation of inventories of purchased and semi-finished parts (Forza and Salvador 2002b). According to Forza and Salvador (2002b), this problem has been tackled by creating modular product architectures, by increasing production flexibility and by improving the production control and planning process, but as the rationalization in other departments, the order process should also be upgraded in order to deliver the generated variety to the markets.

Pulkkinen and Bongulielmi (2004) argue that while different types of modeling methods are available for configuration knowledge, the configuration tool should be in accordance with the organizations that develop, use and maintain the tool while also the product structuring systematics should be considered. Thus, while the organization is affected by configuration knowledge, the modeling of configuration knowledge should be related to the products and it should also be easy to use. They also see that the modeling method can be made less expressive by simplifying the relations between elements. Pulkkinen and Bongulielmi (2004) argue that this will lead to inconsistent configuration models that cannot be maintained. Thus, the organization is affected by the modeling method that provides the configuration models to be used in subsequent phases. While the modeling method affects the organization, also the information exchange between engineering, sales and production is emphasized in Pulkkinen and Bongulielmi (2004). They see that the sales process is interested in the varied proportion of the product, the production system needs to produce the product individual while the engineering provides the documentation needed for the stakeholders. Thus, the information exchange is critical for the configuration process to work properly. Pulkkinen and Bongulielmi (2004) see that information exchange between sales and engineering is usually the most problematic, thus the change processes for products are becoming more important.

2.3.5. Closing aspects of configuration

Product configuration is one way of handling the existing complexity surfacing from the products struggling to satisfy the varying customer needs. A configuration process enables an automatic conversion of customer specifications into the required knowledge used by the subsequent phases.

Configurable products can be considered to be non-modular or modular. Many researchers consider that modularity is needed for the basis of configurable products. Types of modularity affect the entire organization including the configuration maintenance. As the complexity of the products increases, the need for well-defined configuration knowledge is required.

Configuration models present all the knowledge needed in order to execute the configuration process. If the product is modularized and only predefined modules are used for configuration, the task of establishing the configuration knowledge is simplified to the problem of modeling the products and maintaining this knowledge. Maintaining and documenting configuration knowledge is imperative for configurators to work. If configuration knowledge is not maintained, the configurators will be outdated and incapable of producing valid configurations.

2.4. Matrix representations

Matrix presentations have been used widely in the area of modular product design. Most frequently the matrices have been concerned with issues such as platform development and modularity of the product structure. In literature there are many methods that include matrix representations such as MPA (Modular Product Architecture) (Dahmus et al. 2001), QFD (Erixon 1998), MFD (Erixon 1998), K- and V-Matrix method (Bongulielmi et al. 2001), DSM (Design Structure Matrix) (Steward 1981, Browning 1998, Eppinger et al. 1994) and Generic product design process (Ulrich and Eppinger 2000). MPA, QFD, MFD, K- and V-matrix method and the generic process for product development are concerned with modeling the product structure and issues like commonality and variability of the product family. The DSM method is used to analyze and organize the product development tasks in the case of complex development projects. MPA, QFD, MFD and the generic process for product development have been discussed already in section 2.2.4. This section is mainly considering the K- and V-Matrix method and the DSM method.

Gershenson et al. (2004) provide a throughout description of available modularization methods and the related matrix-based methods for presenting modularity. They show that matrix-based methods are widely used to support modularization efforts during the product design process through representing different types of relationships between components. This research is concerned with representing relationships related to components used to document configuration knowledge.

2.4.1. DSM-approach

Steward (1981) formalized the definition of the DSM to be Design Structure Matrix. Steward (1981) concentrated on the aspect of product development tasks and their dependencies. According to Steward (1981), product development is concerned with determining variables which together define the product, how it is made, and how it behaves. All the variables need to be defined, but they have dependencies between each other that determine the order of the definition of the variables. The idea is that before some variables can be determined, other variables must be known or assumed (Steward 1981). This means that the variables have a sequence and also the tasks determining the variables are according to the same sequence.

Browning (1998) also uses the ideas of the DSM approach to understand the iteration needed and its impact on cycle times during the product development process. Browning (1998) uses the definition of DSM to be dependency structure matrix which is a more general name still retaining the DSM from Steward (1981). Browning (1998) defines the DSM matrix as a square matrix with one row and column per activity. As the tasks are placed in roughly chronological order, the networks of the matrices are defined as shown in Figure 16 (Eppinger et al. 1994).

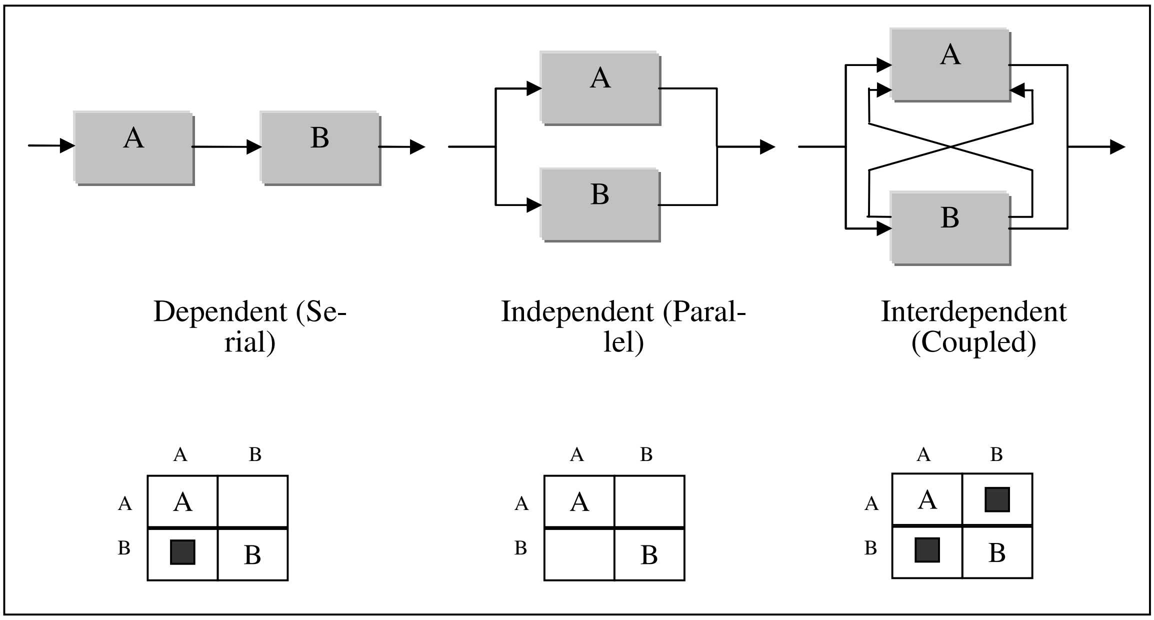


Figure 16. The networks for DSM-matrices (Eppinger et al. 1994)

Eppinger et al. (1994) offer an easy to understand example of the serial, parallel and coupled tasks. If task A is the product design function and task B is the manufacturing engineering function then (Figure 16):

- Serial task could equal the "throw the design over the wall" methodology
- Parallel task could equal unrealistic simultaneous design where A and B are given the same challenge and the product and process is concurrently developed without complex interaction
- Coupled task could equal simultaneous engineering where information exchange is needed and iteration is typical

Considering the lower part of Figure 16, the clarification of the presentation is also given by Browning (1998). As the tasks are placed in timely order, the subdiagonal tasks define the feed-forward information, the superdiagonal tasks specify the need for iteration and rework, and if both of the entries are filled, the task is coupled and heavy iteration can take place and excessive change of information is needed (Browning 1998). Coupled tasks indicate the most complex situation whereas the serial tasks represent the simplest situation where no iteration is needed in the context of connected tasks leaving the parallel situation to be the case where the tasks are not connected by any means. According to Steward (1981), the variables could be determined one at a time if the matrix could be arranged so that it could be in the lower triangular, i.e. no marks would be left above the diagonal. According to Steward (1981), if the elements of the matrix are being rearranged, the same rearrangement needs to be done for rows and columns in order to keep the matrix organized. Even if the reordering of the matrices can be accomplished, the lower triangular matrices are almost impossible to achieve (Steward 1981). Using partitioning of the elements they can be arranged into lower triangular or as close as possible to the diagonal. The formations of smaller square matrices located in the diagonal represent the smallest group of elements that needs to be defined together (Steward 1981). These smaller entities include such elements that have dependencies with each other and cannot be determined alone, i.e. they present the need for iteration between the elements.

Eppinger et al. (1994) have studied the DSM approach in many industrial cases and developed the model further. They have added the aspect of concurrent engineering on the matrix presentation and also added some measurements to define the degree of dependence between the tasks. Eppinger et al. (1994) divide the concurrent engineering into concurrent engineering in small and concurrent engineering in large. The former type of concurrent engineering uses the team integration approach where five to ten people work closely together. The latter type of concurrent engineering describes projects involving hundreds of people and many smaller projects inside the larger one. Eppinger et al. (1994) also discuss the effects of concurrent engineering on the product design projects. Considering the DSM matrices the coupling of the tasks is needed to achieve concurrent engineering, i.e. the representatives of all the functions of the company would perform the design tasks simultaneously and the feedback needed forms the basis for iteration. Concurrent engineering means higher quality in design, but can cause longer product development times because of the iteration needed. The opposite strategy is to speed up the projects by decoupling the tasks. This means less iteration and faster development projects, but the quality can suffer. This means that a compromise is needed in order to optimize the trade-off between design time and quality (Eppinger et al. 1994).

2.4.2. Classification of matrix presentations

Malmqvist (2002) defines the matrix-based product modeling method to be a presentation of some view of the product structure. Malmqvist uses the expression P-DSM (Product modeling Design Structure Matrix) due to the similarity of the above presented DSM method with the exception of concentrating only on the product view. The matrices can be divided into inter-domain and intra-domain matrices (Malmqvist 2002). The intra-domain matrices include the same elements in the rows and columns. In this type of matrix presentation the relations and/or dependencies of the same type of elements are presented. Inter-domain matrices have different types of elements in the rows and columns and the relationships between different element types are presented (Malmqvist 2002). This means that even if the intra-domain matrices include the same elements in the rows and columns, the elements are of the same kind such as modules, parts, functions and design tasks. Then again the inter-domain matrices have different types of elements in the rows and columns and they can be used to analyze a shift from a more abstract element to a more concrete one (Malmqvist 2002).

Malmqvist (2002) divides the P-DSMs into element level and product-level P-DSMs. In addition to this, there are also matrix methodologies that utilize some set of P-DSMs in a systematic way (Malmqvist 2002). Element-level P-DSMs are used to represent the dependencies between element, parts or components in a single product whereas the product-level P-DSMs are used to represent the dependencies between a set of properties and a number of alternatives (Malmqvist 2002). The difference is that the element-level P-DSMs are used for a single product whereas the product-level P-DSMs are used for the entire product family or platforms. The parts of the product-level P-DSMs are the variants of the product level, i.e. the product level P-DSMs are a level higher and the level of decision is altered. Malmqvist (2002) has noticed that the element-level P-DSMs can be divided into the inter-domain and intra-domain P-DSMs, while the product-level P-DSMs only have intra-domain P-DSMs. The classification of matrix-based product modeling method types is shown in Figure 17 (Malmqvist 2002).

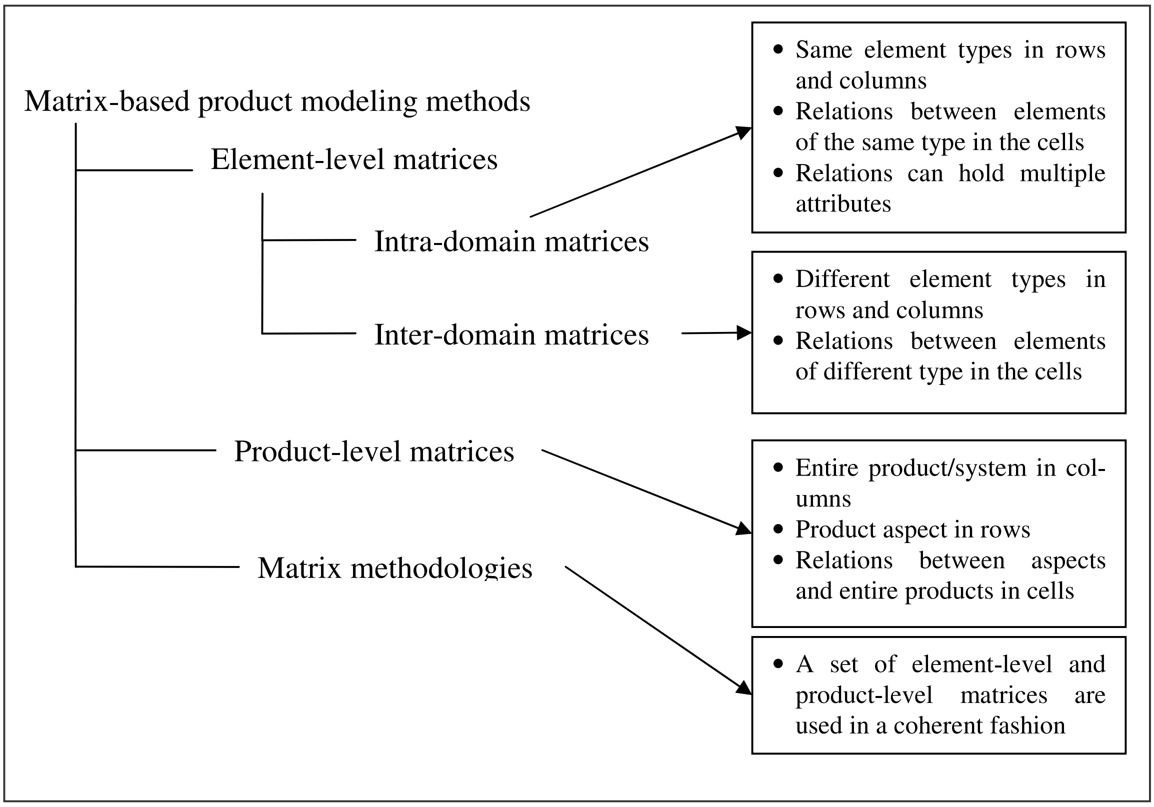


Figure 17. Classification of matrix-based product modeling methods (Malmqvist 2002)

2.4.3. K- and V-matrix Method

Bongulielmi et al. (2001) have developed a matrix-based method to control external and internal variety of the product. The main purpose has been to establish a well-defined description language for variant products to satisfy the needs of the configurator. The knowledge developed during the product realization process is gathered to be used in subsequent phases of the product's life cycle (Bongulielmi et al. 2001). The aim of Bongulielmi et al.

(2001) is to manage and analyze the variants of the product and also to reflect the logic of the product structure in a systematic way to marketing and sales. The main driver is the usability of the description language to the industrial cases, i.e. the description language needs to support the variant generation, description and management during the product realization process and later during the sales process (Bongulielmi et al. 2001).

The method presented by Bongulielmi et al. (2001) consists of two types of matrices. The first type of matrix is called the K-Matrix which is used for mapping two correlated product views. This matrix has two different views as shown in Figure 18. The number of views is unlimited and the company specific views can be established (Bongulielmi et al. 2001). According to Bongulielmi et al. (2001), the K-Matrix can be used to manage the product variety, both internal and external. The idea is that the customer view is translated into the technical view (Figure 18) and the effect of customer variants can be reflected on the technical side.

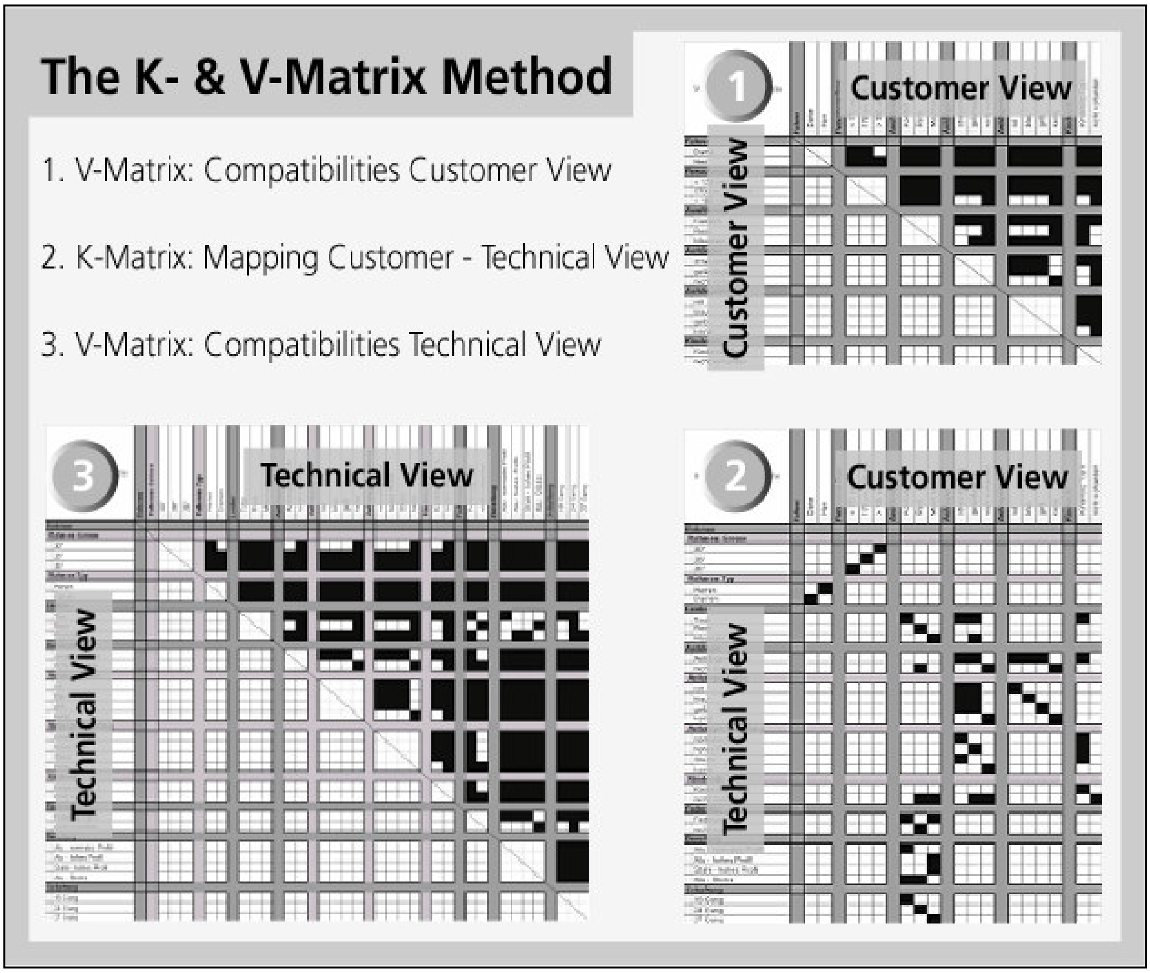


Figure 18. The K- and V-Matrix method (Bongulielmi 2002)

The V-Matrix concentrates on one view and the rows and columns have the same data in the same order, i.e. the data of the rows have been transposed into the columns. The matrix is a square matrix showing the dependencies between the selected elements. Because the type of matrix is a square one, there is no need to present the dependencies at the lower triangular of the matrix as shown in Figure 18. According to Bongulielmi et al. (2001), the V-Matrix can be used to compare all the variants of every module with each other to define all the combinations in the product family. If the module level is shown in the V-Matrix, the compatibilities between the modules are shown in the cell. Considering the three matrices needed there is only need to construct two and the third can be derived from the first two (Bongulielmi et al. 2001). According to Bongulielmi et al. (2001), the V-Matrix for customer view can be established by the knowledge provided by the K-Matrix and the V-Matrix for the technical view.

Considering there is a total of three matrices, two of which show the compatibility of the elements and one integrates the two views together, a well-defined software tool for interpreting the matrices is needed. The manual configuration of the product can be confusing in the case of multiple matrices. Bongulielmi et al. (2001) have also developed software system that is used to edit, maintain and query the data from the databases serving the methodology. This enables the effective use of the K- and V-Matrix method to be used in industrial environments.

Bongulielmi et al. (2002) discuss the relationship between the K- and V-Matrix method and other matrix presentations found in literature and also the position of the methods during the product design process. Their idea is that the configuration knowledge should be generated during the detail design as part of the product design process. At this point the aspect of configurability is considered and the product structure finalized. The methods compared were QFD (Bongulielmi et al. 2002), MFD (Erixon 1998), MPA (Dahmus et al. 2001), DSM (Steward 1981) and DfV (Martin and Ishii 2000). Bongulielmi et al. (2002) use the classification presented by Malmqvist (2002) and locate the methods as follows (Table 3):

Table 3. The classification of matrix presentations (Bongulielmi et al. 2002)

Inter-domain matrices	Intra-domain matrices
QFD	QFD-roof
MFD	DSM
MPA	DfV
K-Matrix	V-Matrix

According to Bongulielmi et al. (2002), the main difference between the methods presented (Table 3) and the K- and V-Matrix method is that the values of the cells in the K- and V-Matrix presentation are either “0” or “1”, which means that there exists a relationship or there is no relationship between the two elements. The second difference is that the K- and V-Matrices are set up during the late phases of the product design process (Bongulielmi et al. 2002). Bongulielmi et al. (2002) consider all the other methods to support the design teams in issues related to solving product architecture problems like commonality and modularization. The K- and V-Matrix method is not used for the above mentioned product structure issues, but is used to solve problems related to product configuration related issues. According to Bongulielmi et al. (2002), the K- and V-Matrices can be set up after the product structure has been established and the general design of the modules is done. The main point that Bongulielmi et al. (2002) present is the fact that the variant modules (the technical view) are defined during the design of the product structure and the functional view (customer view) related options are defined during the planning phases of the innovation process (see the general process for product design, Ulrich and Eppinger 2000, section 2.2.4). This way the customer needs have been considered. The product structure established can be described as matrix presentation to be used during the subsequent phases of the product design process and also during the configuration process. The benefits and tools for analyses that the K- and V-Matrix method provides for the design team are as follows (Bongulielmi et al. 2002):

- The overview of the degree of fulfilling the customer requirements (K-matrix)
- The overview of the knowledge volume (number of rules and constraints) due to exceptions and sub-optimal product family structure (V-matrix)

The role of the K- and V-Matrix method is to provide the tools for handling the configuration knowledge and to be a complementary tool for other matrix methods. The combination of the K- and V-Matrix method with other matrix methods supports (Bongulielmi et al. 2002):

- The design of modular product architecture
- The consideration of aspects concerning the configuration during the design process
- A systematic description of a major part of the configuration knowledge
- A communicative bridge between the engineering and the sales department

Next to Bongulielmi (2002), Aldanondo et al. (2000) use matrices as a basis for their expert configurator. This type of configurator is used for highly customized products and the need for expressing the configuration knowledge is even more important since usually the routings and related cost estimations are also influenced. Aldanondo et al. (2000) use a graphic model next to the matrices, but as the dependencies get more complex, the matrices are used as the primary tools for presenting dependencies. They use matrices for presenting the dependencies between (Aldanondo et al. 2000):

- Two attributes
- An attribute and a component characteristic
- An attribute and a quotation characteristic
- An attribute and an operation characteristic

As the above list is considered, Aldanondo et al. (2000) cover the function model, the bill of material domain, the quotation domain and the routing domain with matrices.

2.4.4. Closing matrix presentations

Matrix methods have been used to simplify various different problems over the time. The DSM approach has been used to understand the dependencies of the product development process and the need for iteration of the tasks during the development process. The ideas of the DSM approach will be used in this thesis when defining the basis for the configuration matrices.

As noticed the K- and V-Matrix method is concerned with the problems related to maintaining configuration knowledge. The usability of the matrix presentation in this area simplifies enormously the complex product structures. The K- and V-Matrix method integrates the technical and customer view into one matrix, the K-matrix, which is used to show the dependencies between two different aspects, i.e. the rows include one aspect and the columns include the other. The K- and V-Matrix method has a straight relationship between the ideas of this thesis. The purpose is to give a more production-based view into the design and marketing processes. The ideas of the configuration management are the same and more integration between the processes and functions is established.

Pulkkinen et al. (2004) consider the K- and V-matrices to be able to present one-to-one mapping between properties and components and more sophisticated models are different kinds of object models. They also see that usually the product structure is not so modular that representing only the part selections is enough to present the configuration knowledge. For their case studies the simple approaches, such as matrix approaches, delivered satisfactory results even if complex methods were not used. Pulkkinen et al. (2004) conclude: "… selecting the context of configuration, the phase of sales-delivery process and subset of the product family, appears to be more beneficial to the immediate application than the modeling conceptualization.", i.e. industrial cases try to find an appropriate approach to configuration knowledge management whereas scientists are trying to develop modeling concepts and methods.

2.5. Producibility of the product family

According to Torvinen et al. (2003), the concept of producibility of a product family includes the aspects of manufacturability, assemblability and many additional aspects. According to their definition, producibility of a product includes the following aspects:

- Purchaseability of materials, components, parts and modules
- Material's ability to be manufactured, formed etc.
- Manufacturability of a part, module and product
- Assemblability of a product, module or subassembly
- Configurability and modularity
- Measurability
- Testability
- Inspectability
- Packageability
- Installability
- Disassemblability of a product, module or subassembly
- Recycleability or a product, module or subassembly

While companies are facing increasing challenges to satisfy customer needs, designing for producibility offers means to avoid disturbance due to the customer specific, configured products (Torvinen et al. 2004). The idea is to integrate production and product design to be able to offer enough product variants and at the same time to make efficient production of configured products possible. This thesis is considering producibility through modularity and configurability. While Torvinen et al. (2003) concentrate on developing products that can be efficiently produced, Lapinleimu offers an idea of an ideal product for an ideal production (Figure 19).

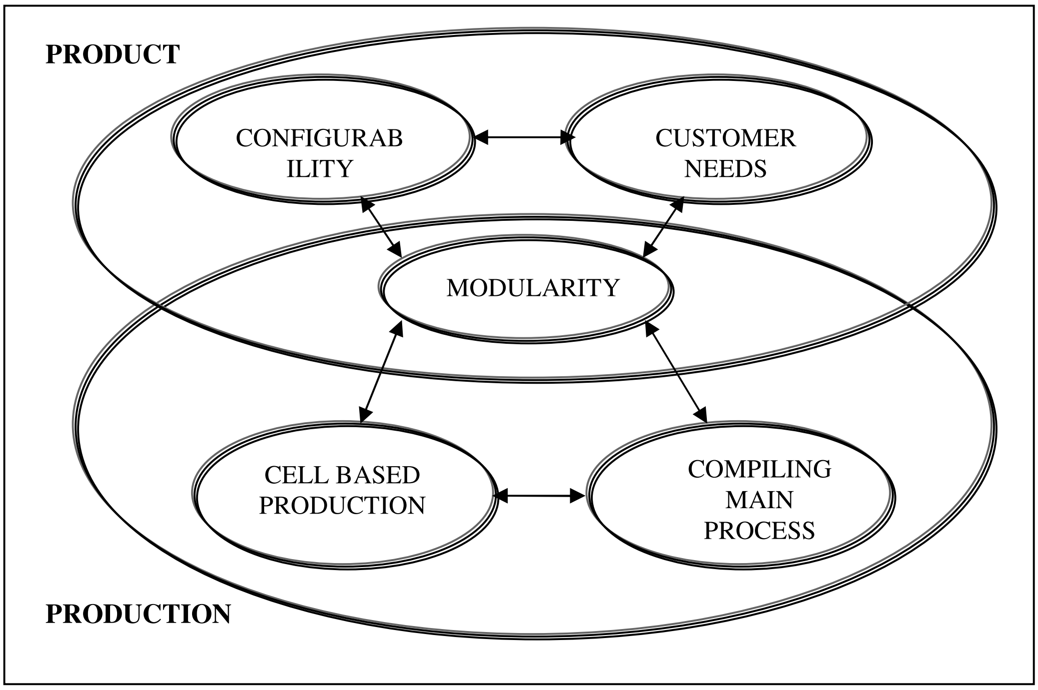


Figure 19. An ideal product for an ideal production (Lapinleimu 2000)

As shown in Figure 19, modularity acts as the center for a product as well as for the production system, i.e. modularity offers means to use module-based production. In Figure 19 the idea that there needs to be enough similarity between the product structure and production system, while simultaneously enabling configurable products, is clarified.

This thesis is considering producibility in the context of configurability and modularity. As mentioned before, the aim is to provide a modular product that satisfies the requirements of production and sales simultaneously. Thus, configurability and similarity between production and product structure needs to be solved. This provides a producible product structure. When considering the evolution model presented by Lehtonen (2003), producibility can be seen in relation to configurability and modularity.

Torvinen et al. (2004) see that poor producibility will increase the instability of the system. Producibility is considered to be the property of a product supporting production system (Lapinleimu 2000). Thus, producibility of the product is offered to provide minimal amount of deviation to the production system caused by the product itself. According to Torvinen et al. (2004), "…new product drawings, new instructions, new tooling etc. can cause instability. Everything that is new requires a learning process that 'disturbs' the system…" The integration between production and product development becomes important. Jensen and Hildre (2004) argue that there are shortcomings in modularization methods when considering manufacturing. The concept of producibility and model provided by Torvinen et al. (2004) provides insights into this area holistically. Torvinen et al. (2004) present the CSD model (Collective System Design) to provide a holistical view of production system and the importance of product to the production system simultaneously. For example, producibility is heavily considered by Torvinen et al. (2004) because the system needs predictable outputs and producible products provide these outputs.

Torvinen et al. (2004) consider product design and production system in a way that product design provides a product that supports the production system in a way that stability is reached. Torvinen et al. (2004) conclude: "Product design for producibility encompasses topics such as architectural design, detailed design (including DFMA (Design for Manufacturing and Assembly)), process and flow planning, and operations planning. When the foundation for the stable manufacturing system has been laid with the help of highly producible products, the goal of the CSD framework proceeds with ensuring the existence of standardized high quality work to minimize all undesired deviations…". Thus, they provide a well defined framework for integrating product design with production.

Delayed differentiation (Ulrich and Eppinger 2000, Selladurai 2004, Partanen and Haapasalo 2004) can also be considered as part of producibility. Modularity offers means for delayed differentiation when designed properly into the product. Delayed differentiation and mass customization are often related. Partanen and Haapasalo (2004) see that requirement for rapid response manufacturing (Suri 1998) and fast delivery of configured products can be satisfied with modularity and standardized methods. Also Selladurai (2004) sees that process standardization enables mass customization, thus modularity provides means to offer pull systems (Pine 1993, Torvinen 2004) next to the use of push systems (MRP, Material Requirement Planning) through delayed differentiation. The types of modules in modular systems either enable or disable the use of delayed differentiation.

2.5.1. Closing producibility of the product family

This thesis is concentrating on producibility of a product family especially using the aspects of configurability and modularity of product structures. Increasing needs to satisfy customer needs affect the entire organization while configurable products are one way to provide customer specific product individuals. Modularity is often offered as means to handle complexity and to enable efficient use of configurable products. Simultaneously with configurability there are needs to have producible products, i.e. the product structures should be in accordance with the production system. Configurability and similarity between production system and product structure represent usually different types of modularity, i.e. functional or structural modularity respectively.

Connecting these ideas to the evolution model of modularity (Lehtonen 2003) and configuration matrices, a systematic way of developing modularity can be established. Considering producibility, it is possible to see modularity as means to develop producible products while ensuring the aspect of configurability, i.e. modularity needs to be seen broadly in order to eliminate sub-optimizing the product structure.

3. CONFIGURATION MATRICES AND THE IMPLICATIONS TO ORGANIZATION – THE DEVELOPED METHOD

3.1. Introduction

The main purpose of this research is to build a method to enable a systematic configuration process by establishing the configuration models for the products. Modularity is imperative in complex products to make the configuration process a routine task during the order-delivery process. When modules and the modularity are well established and the change management is under control, the configuration process can be effective.

The real life examples give good understanding of the requirements that need to be taken into a consideration when deciding what type of configurator to be used in a certain situation. Configurable products and their configuration knowledge tend to change as the lifecycle of the product advances. The reason is the changing customer needs that must be satisfied. Modularity and configurability offer means to provide variety, while one of the main issues to deal with is the configuration knowledge, its modeling and documentation. As shown in this chapter, careful management of configuration knowledge gives many advantages when a configurable product is the case.

This chapter provides insights into the developed method for maintaining and creating configuration knowledge. Section 3.2 provides the framework for this thesis. In section 3.3 the different features of the matrices are presented and in section 3.4 the process of developing the matrices is introduced. Section 3.5 discusses about the role of product structure briefly while section 3.6 considers the evolution of modularity and the configuration matrices. Section 3.7 concentrates on configuration matrices in changing environment and section 3.8 presents the configurator built on top of the configuration matrices. Finally in section 3.9 the conclusions are made concerning this chapter.

3.2. The framework for configuration matrices

Configuration matrices are provided to structure, model and document configuration knowledge dispersed into the organization. By documenting this knowledge the grounds for an effective configuration process are established. One of the main issues to notice is the integrative factor of configuration knowledge to the organization. When considering the case of configurable products, modularity is often related to the topic. Thus, modularity, configurability and configuration models are related. When considering modularity and configurability, the company must define module interfaces so that the requirements for configurability are satisfied. It is the capability of the company to design the product architectures so that the similarity between product structure and production system and configurability are simultaneously reached. To make this possible the documentation of configuration knowledge and the understanding of modularity must be in high level, i.e. it depends on the capabilities of the company. Sub-optimization is very easily accomplished for both modularity and configurability while holistically designed products provide the best benefits for the company.

Strategic issues such as modularity and configurability are supplemented by configuration matrices to provide the grounds for systematic processes. Considering the next level, generic product structures provide means to handle different situations in the operative level. The generic product structures are provided by the configuration matrices and tools related to operational level are established by information technologies. Figures 20 and 21 provide all the tools related to configuration matrices developed during this research. The strategic level issues presented above with the operative level tools provide the framework for this research.

In Figures 20 and 21 all the tools related to configuration matrices have been located into the organization using the matrix representation of the configuration knowledge. Figures 20 and 21 show the meaningful functions in the context of configuration matrices and plot the tools as well as inputs from different stakeholders into the matrix. For all the stakeholders there are some tools used when considering other stakeholders. Horizontally interpreting Figures 20 and 21 concentrate on one stakeholder compared to others. Inputs are derived from the stakeholders in the columns to affect the stakeholders in the rows. There are no inputs for the diagonal of the presented matrix, but the tools can be considered in the same context, i.e. this is the part where, for example, production based tools are considered in the context of production. Thus, the diagonal of the matrix presents the tools that are used by the stakeholder. The gray areas have not been considered deeply in this thesis, while the light blue areas and the diagonal (orange) have been covered.

STAKEHOLDER	CUSTOMER	MARKETING/SALES	PRODUCT DEVELOPMENT
CUSTOMER	Tools: -N/A	Tools: -N/A Inputs: -Marketing configuration rules -Guidance	Tools: -N/A Inputs: -Mass customized products -Products to meet customer needs
MARKETING/SALES	Tools: -Configurator -Validity checks -Change processes -Price inflation -Option frequency Inputs: -Customer specifications/changes -Volume options -New requirements	Tools: -Marketing configurator -Option frequency analysis -Pricing software -Change processes -Validity checks -Price inflation analysis	Tools: -Pricing software -Change processes -Validity checks Inputs: -Generic product structures -Configuration rules
PRODUCT DEVELOPMENT	Tools: -Option frequency Inputs: -Customer needs -Feedback from products -New requirements	Tools: -Platform analysis Inputs: -Customer needs -Feedback from products	Tools: -Plafform analysis -Commonality of components, modules -Base machine analysis -Modularity development based on matrices -Change processes
PURCHASING	Tools: -N/A Inputs: -Sold products	Tools: -N/A Inputs: -Sold products -Product individual structures -Forecast machines	Tools: -N/A Inputs: -Generic product structures
PRODUCTION	Tools: -Configurator Inputs: -Customer specification	Tools: -Configurator Inputs: -Customer specification -Delivery date	Tools: -Commonality of components, modules -Base machine analysis -Cell re-engineering Inputs: -Generic product structures -Module interfaces
AFTER SALES	Tools: -N/A Inputs: -Customer needs for the life cycle	Tools: -N/A Inputs: -Customer satisfaction	Tools: -N/A Inputs: -Life cycle support

Figure 20. Tools related to configuration matrices

STAKEHOLDER	PURCHASING	PRODUCTION	AFTER SALES
CUSTOMER	Tools: -N/A Inputs: -Contributes through production	Tools: -N/A Inputs: -Produced product individual	Tools: -N/A Inputs: -Maintenance of the configured product individual during the life cycle -Customer service
MARKETING/SALES	Tools: -Pricing software Inputs: -Component prices -Lead times	Tools: -Pricing software Inputs: -Labor costs -Lead times	Tools: -N/A Inputs: -Life cycle costs
PRODUCT DEVELOPMENT	Tools: -Commonality of components, modules -Base machine analysis -Platform analysis Inputs: -Component prices -Suppliers -Lead times	Tools: -Commonality of components, modules -Base machine analysis -Plafform analysis Inputs: -Production system -Produceability -Product changes for produceability	Tools: -N/A Inputs: -Feedback from products -New requirements
PURCHASING	Tools: -Active module based analysis -Through put time analysis -Invalid parts analysis	Tools: -Active module based analysis -Through put time analysis Inputs: -Volume components and modules -Consumption of parts	Tools: -N/A Inputs: -Spare part orders
PRODUCTION	Tools: -Active module based analysis Inputs: -Supply dates	Tools: -Through put time analysis -Invalid parts analysis -Cell re-engineering -Platform analysis -Product kill analysis -Base machine analysis -Change processes -Production configurator -Validity checks	Tools: -N/A Inputs: -Spare part orders
AFTER SALES	Tools: -N/A Inputs: -Spare part issues	Tools: -N/A Inputs: -Spare part issues	Tools: -N/A

Figure 21. Tools related to configuration matrices (continued)

Configuration knowledge maintenance is very essential for configurable products and it could be part of Figures 20 and 21. Tools used by configuration knowledge maintenance are naturally heavily related to configuration matrices. Configuration matrices themselves are a tool that configuration knowledge maintenance uses. Configuration knowledge maintenance is usually situated under product development or production. The main point for Figures 20 and 21 is to provide understanding how configuration matrices affect its surroundings rather than concentrate on using the matrices. These issues will be dealt with later in this chapter when configurator and configuration knowledge maintenance are considered. Figures 20 and 21 have been developed during the case study considering all the stakeholders separately in the context of configuration knowledge and configuration matrices established during this thesis.

3.3. Different features of the configuration matrices

Configuration matrices have been built as light as possible in order to generate all the needed information as simple as possible (Nummela 2004). Configuration matrices are designed to help the organization to understand the product structure and to reveal the usually hidden knowledge that is critical in order to effectively produce configurations and also to maintain the knowledge related to configurable products. It is noteworthy that the matrices can present the knowledge to be used both for manually or automatically controlled configuration processes as they are formed to support the configuration task generally. The main purpose for the configuration matrices is to present all the options saleable at the moment and the connections between the combination of the selected options and the modules for production, i.e. the configuration rules are established by revealing the configuration related knowledge. This means that the generic structure for options is needed and also the generic product structures are imperative to have.

As presented by Lehtonen et al. (2003) the second phase of modularity is functional and the third phase of modularity is platform based. The idea of configuration matrices is that there is a connection between the modularity of the product structure and the matrices so that the matrices can be used to guide the product development to develop new types of platforms and types of modularity systematically as the knowledge of the current situation of the product structure and its modularity increases. The power of the configuration matrices is realized when integrating the generic product structures to local ERP (Enterprise Resource Planning) or PDM (Product Data Management) systems. Firstly, this enables the automatic transfer of the configuration rules into the system and the tedious manual work is eliminated. Secondly, the integration enables the building for various pieces of software to analyze the structure and the current situation in production, marketing and design department (Nummela 2004).

3.3.1. The structure of the configuration matrices

Configuration matrices are created so that the features and their options are connected to the module level of the product structure. Configuration matrices are element-level and intra-domain P-DSMs according to classification given by Malmqvist (2002), i.e. the configuration matrices are square matrices with the same elements in rows and columns (see section 2.4.2). The elements found in the matrix are all the modules and the features including their options to choose from. This implies that the configuration matrices include simultaneously two different types of elements located sequentially in the rows and columns of the matrices. If there is a need for making the selection of a module by a combination of different options, the module is located after the last option selection that affects the module selection. As the matrices are formed, the features and their options should be generic in nature since all the saleable options should be predefined to be used in various parts of the organization. Also the generic product structure starts to appear as the matrices are formulated. The generic product structure for a specific product family is the collection of active and in use modules presented by the configuration matrices. The generic product structure defines all the possible modules that can be part of the configurations of a specific product family. As this generic product structure presents all the possible product variants, the impact of managing these structures generates a powerful tool for different stakeholders of the company. As shown in Figure 22, the collection of options, all the modules and the base machine is presented for an example configuration matrix.

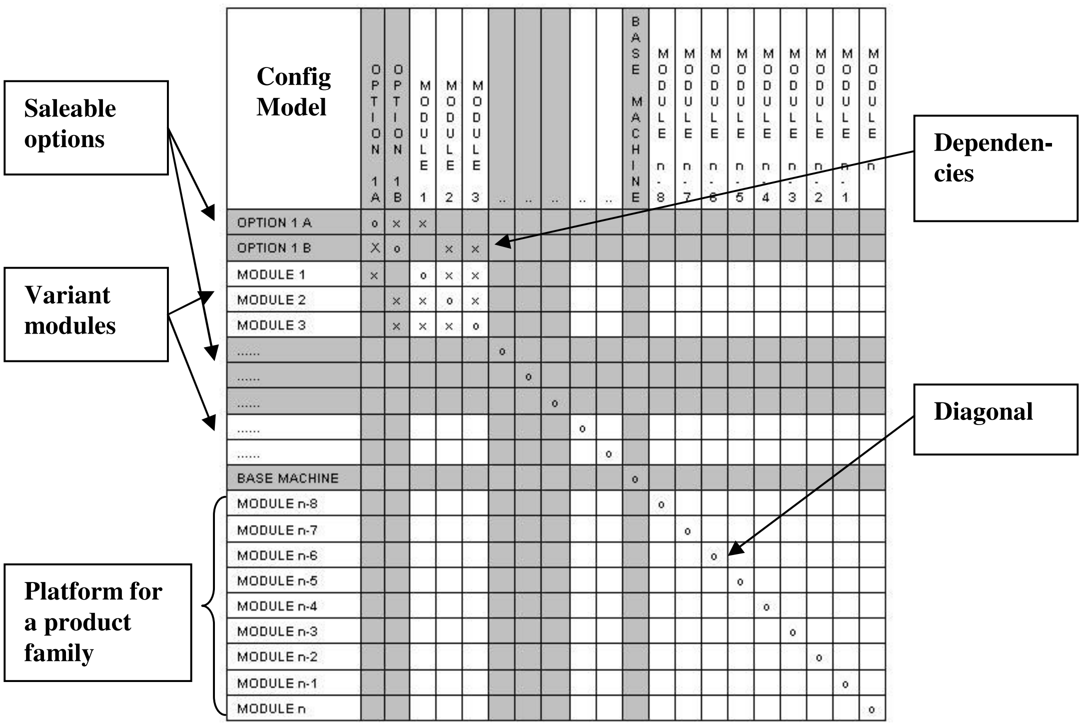


Figure 22. Configuration matrix

As the configuration matrix representation is used for a specific product family, the modular system presented by Stake (1999) can be used to analyze the elements and features of the matrix representation (see 2.2.3). As a modular system includes basic modules (platform) and variant modules to generate product individuals by configuring the customer products, the structure of the configuration matrices includes simultaneously the base machine (platform for a specific product family), all the variant modules and also the options to configure the product individuals. Together with the variant modules, the base machine forms the generic product structure for a modular system.

Configuration matrices have been developed mainly to satisfy the needs of the configurator which means that maintainability is needed for the matrices. As changes take place the matrices are updated and the configuration knowledge is maintained in a systematic way.

The visualization of the product structure can be used to understand the modularity of a product more deeply. As the evolution model (Figure 14) presented by Lehtonen et al. (2003) is considered, the product structure and its phases can be connected to company strategies. As the surroundings of the company alter, there is need to reconsider the product structure and its modularity as well. Holmqvist and Persson (2004) conclude that product modularity is not only a product issue, i.e. modularity should be considered holistically in order to minimize sub-optimization. They see that modularity affects the product, design processes and also the organization. As shown in the case study of this research, the ideas of Holmqvist and Persson (2004) can easily be agreed upon. Configuration matrices reveal the knowledge underlying the products of the company in a way that the structures can be analyzed and actions taken to consider the wanted modularity. As shown in the case study, the very problematic area of understanding the modularity of the products can be understood by using configuration matrices and the next level of modularity can be reached.

3.3.2. Configuration rules

Configuration matrices use AND and OR operators to express the needed dependencies. As shown in Figure 22, the dependencies are established by symbol "x" in configuration matrices. The same symbol is used both for marketing rules and production configuration rules. "o" represents the diagonal (Figure 22).

As shown in Figure 22, the configuration rules that can be identified from the matrix representation are marketing configuration rules and production configuration rules. Marketing configuration rules control the marketing configuration task to enable the salesman to select valid options, i.e. the compatibility of the selections is secured. As the options of different features are presented in gray color (Figure 22), the marketing rules appear in the crossing areas of the options. In order to present the rules and the matrices as light as possible, the identification of marketing dependencies is marked only when the marketing rules are apparent and some of the options are either possible or not possible due to the dependencies between other options. These marketing rules can also be used to control the variations offered to the markets, i.e. the options can be easily made possible or not possible by adding the rules manually.

Production configuration rules are established by showing the dependencies between modules and corresponding options. These production configuration rules consist of either one option as the algorithm to select modules or a set of options to form a combination of the options to select proper modules to form the product individual. The matrices are formed so that the module level of the product is located under the options connected to different modules. As the sequence of the option selection is formed, the production configuration rules are combinations of the selected options. The sequence (also an issue with expert configurators as presented by Aldanondo et al. (2000)) of the matrix representation is imperative since some rules are needed in order to maintain the configuration knowledge effectively. If the sequence of the accomplished matrix representation is altered, all the dependencies still remain, but they can be located in different places which makes the interpretation of the representation difficult and the development of software tools impossible. This is mainly due to the dependencies between the options to be selected. There is a lot of flexibility to form the matrices, but a logical process needs to be accomplished to enable an effective formation of the configuration matrices.

As the dependencies between the modules and options as well as dependencies between options grow bigger, the configuration rules are formed more and more complex in nature. As shown in Figure 22, the production configuration rules are formed by considering the option at hand and the options selected before, i.e. the combinations of the options regarding a certain module are established. The selected option might suggest that there are, for example, five modules that are possible to the customer product structure as the former selections make the selection of the module either valid or invalid.

Marketing configuration rules are produced by the idea that some options make other options possible or not possible to choose, i.e. the nature of the marketing rules is a type of making selections available or unavailable. Product configuration rules are shaped so that the operator AND is used in the case of different feature options whereas operator OR is used when the options of the same feature are concerned. In order to formulate exact configuration rules the sequence of the saleable options is critical as well as the location of combinatorial option selections to form a clear representation. The following example will clarify the presented ideas.

3.3.3. Configuration matrices: a general example

This example is based on Nummela (2004). The example has been made easier to read by considering bicycle as an example. A part of configuration matrix is shown in Figure 23.

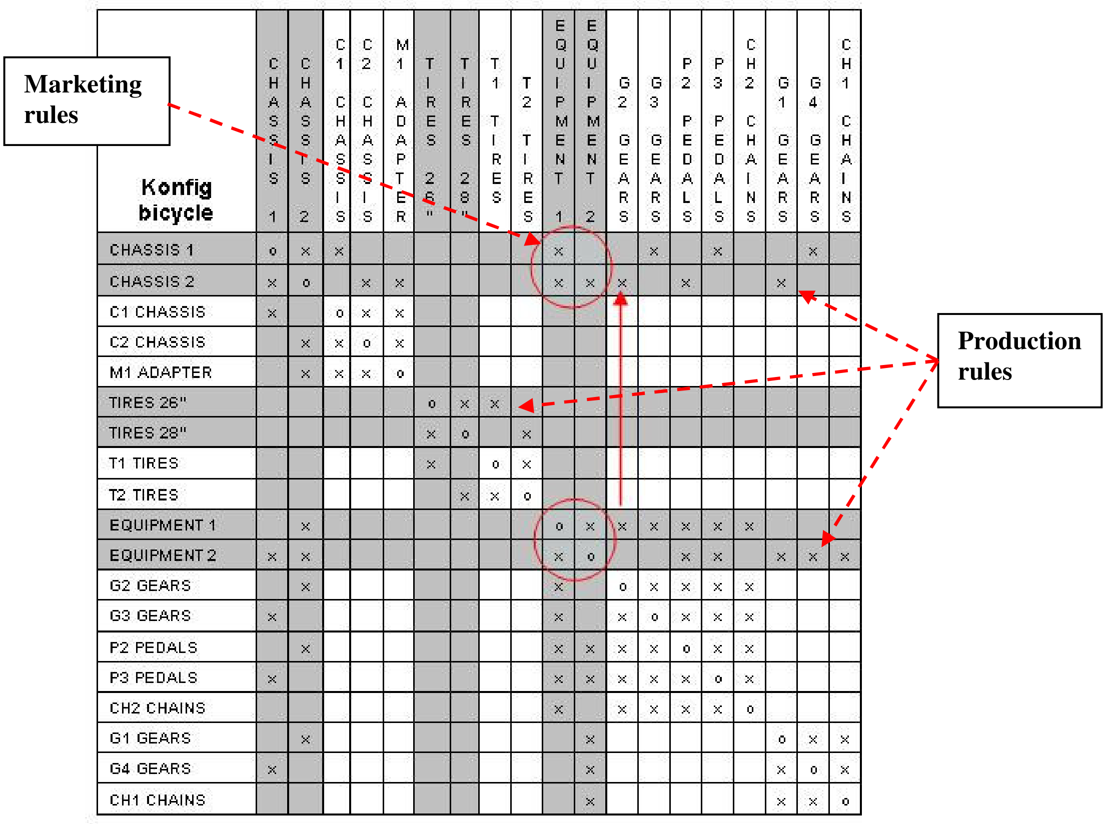


Figure 23. Part of an example configuration matrix

As shown in Figure 23, the gray areas represent customer choices made during the selling process. The module level is the white areas of the matrix, i.e. the modules depending on the choices made by the customer in order to generate customer specific product structures. The purpose of the matrix representation is to present all the possible choices and related modules in order to generate valid and complete configurations. It is obviously imperative to have right kind of product structure, types of modules and modularization in order to make complete configurations and configuration models.

The power of the matrix representation is that all the configuration rules, both production and sales can be presented simultaneously. Considering the configuration process and generation of the customer specific product individuals, the customer choices need to be connected to the module level. This is obvious if the goal is to generate customer structures at the minimum amount of work. The critical part for the generation of the matrix is actually the experience of the personnel building the matrices. The problem is that while the matrix presents all the combinations of the possible modules, the rules can appear in different parts of the matrix. The reason for this is mainly the sequence of the choices present in the matrix.

Considering Figure 23, the sequence of the choices is predetermined in the context of the options in order to generate compatible configurations. If a feature is isolated, i.e. it has no dependencies with other features, there is no difference at what point the selection will be made. Considering routine manual configuration, the building of the configuration will start from the selection of feature 1 (chassis 1 or chassis 2), Figure 23. The example in Figure 23 is concentrated on the selection made concerning equipment package 1 or 2. When arriving to equipment selection, the dependencies between features (sales configurator) can be checked by moving upwards the matrix and studying the possible marks "x" between the features. There are no dependencies between the options found in tires selection and equipment package selection, i.e. the options of tire selection have no dependencies with the options of equipment selection. The tires selection can be shifted considering the sequence of the features because the lack of dependencies between other features, but it can also be at this point. In the example when considering equipment 1 and 2, there are dependencies that form the marketing configuration rules between the options of chassis selection as follows (Table 4):

Table 4. Possible choices (Nummela 2004)

	Chassis 1	**Chasis 2**
Equipment 1	Possible	Possible
Equipment 2	Not possible	Possible

According to Table 4, equipment package 2 is not possible to select if there has been a selection of chassis 1 before. The reason for a clear sequence between the selections of features is imperative since if the chassis selection and equipment package selection would be shifted around, it would be possible to select equipment 2 and chassis 1 to the same configuration. Even if the matrix has a strict sequence between the selections of the features, the configurator can be more flexible since the selection can be done virtually at any sequence since the configuration model holds all the knowledge that is needed and the technology used by the configurator determines the possibilities of the automatic configuration process. These dependencies between the features and their options make the configuration rules for marketing and sales.

Continuing with the example the next phase is to make the decision between selecting equipment 1 or 2 from equipment feature. After the selection the matrix shows all the dependencies between the modules and the options under consideration and options selected in the previous stages. Selecting equipment 1 will suggest the selection of modules G2, G3, P2, P3 and CH2. For example, at the previous stages the selection of chassis 1 and 2 has been made. Considering the example in hand selection of equipment 1 will lead to the selection of the following combination presented in Table 5 when chassis 1 has been selected.

Table 5. Selection of modules (Nummela 2004)

	Selected	Selection string (selected if…)
G2 GEARS	No	equipment 1 AND chassis 2
G3 GEARS	Yes	equipment 1 AND chassis 1
P2 PEDALS	No	equipment 1 AND chassis 2
P3 PEDALS	Yes	equipment 1 AND chasis 1
CH2 CHAINS	Yes	equipment 1 AND (chasis 1 or 2)

Module CH2 CHAINS will be selected into the customer structure every time equipment 1 has been selected. There is no need for the mark "x" when the module will be included at all times considering a specific option. These dependencies cover the configuration rules for creating the customer specific product structures. Considering this example the customer specific product individual consist the following modules (Chassis 1, Tires 26" and equipment 1 selected): C1 CHASSIS, T1 TIRES, G3 GEARS, P3 PEDALS AND CH2 CHAINS.

3.3.4. Integration between the matrices and ERP / PDM systems

As the purpose for configuration matrices is to focus on visibility of the configuration knowledge in order to create maintainable configuration models, the intention is to establish a support system for different stakeholders that are dependent on the knowledge related to configurable products. The knowledge used by the configuration process can be used widely since this kind of knowledge holds the very essential product information in use in a company at a given moment. Considering the structure of the configuration matrices the generic product structure of a specific product family holds all the possible modules that can be used for the product family variants. As the level where the configuration task is executed is defined, the next level in modularized products is usually fixed as the module interfaces are according to configurability of the product at the maximum size of modules, i.e. the level below configurable module level is standard. Thus, the level from where the production configuration rules can be established is the final level that is variable. When integrated with the various databases of the company, the knowledge of the configuration matrices can be used systematically to address the information needs for many stakeholders.

As mentioned before, the most important issue is to handle the configuration rules and the integration with databases enables the support system to automatically feed the configuration system with rules, generic structures, features and their options as well as pricing information. While the configuration rules can be maintained in a supporting system, the manual labor needed to update the configurator is diminished. The next issue is related to the generic structure presented by the configuration matrices. While the active modules are presented in configuration matrices, the PDM / ERP systems hold the fixed structures in their databases, i.e. the active parts and components can be defined at any time using the generic product structures found from the matrices.

The nature of software tools possible to build has three different viewpoints. First, there is the configurator that can be established. This is deeply considered in section 3.8. Second, there are tools that interpret the configuration models and the integration between matrices. The nature of this type of tools is to satisfy the needs of the configuration process in the form of configuration knowledge and to analyze the properties of the configuration matrices and the generic structures within the matrix representation as well as between different matrices. The third type of tools uses the data from the matrices and from the company databases to generate various analyses effectively. The idea is that the data stored in the ERP / PDM databases can be connected to the generic product structures. This connection enables the creation of many tools for analysis to aid the organization and to support decision making.

3.3.5. Techniques used to present configuration matrices

Configuration matrices have been established with Microsoft Excel. Excel has been chosen mainly because the purpose is to present the configuration matrices as simple as possible and to secure the compatibility in Windows surroundings. The configuration matrices are used as databases to be used in various analyses presented by the framework. These analyses have been established by user interfaces programmed with VB (VisualBasic 6.0) and partly with ASP (Active Server Pages) to be used through internet or intranet. The idea is that administrative tools should be done by VB and the most frequently used user interfaces by ASP. This will eliminate the need for updating the programs in many personal computers and the possibility to use company intranet can be effectively used to support configuration matrices. Generally the analyses use company databases and matrices simultaneously to interpret wanted attributes. During this study all the company specific databases were Oracle databases and the connection to the databases has been established by ODBC (Open DataBase Connectivity) drivers which have been used to collect data with SQL (Structured Query Language) -queries.

For the configurator presented in section 3.8, all the databases were established with MySQL-databases.

3.4. The process of creating configuration matrices

Configuration matrices can be generated both for new and existing products. As the product design is responsible for generating configuration knowledge for the rest of the organization, the first process is supposed to tie together the configuration matrices and product design. In many cases it is useful to model the existing products and reveal the hidden configuration knowledge dispersed over the entire organization. Process for modeling existing products is also presented.

3.4.1. Process for generating configuration matrices during product design

Aarnio (2003) ties the MBI method together with the processes presented by Pahl and Beitz (1986) and VDI 2221. MBI integrated with VDI 2221 is presented in Figure 24. Aarnio concentrates heavily on configurability of the product structure while also stating that other needs of the organization need to be considered while generating product structures. As the product structure and the modularity it offers should holistically meet the needs of all the stakeholders of the company, the configuration knowledge needs to be presented effectively. It sounds natural that while the product is being designed, especially for mass customization in mind, the configurability and the knowledge related to product configuration should be documented during the design phase. Even if all the presented design processes can be supported by configuration matrices, the MBI method is selected. This is because MBI is meant to be used with complex mechatronic products and the idea of configurability is essential. Figure 24 shows the MBI method presented by Aarnio (2003) and the formation of configuration matrices are concluded in the results. After the customer needs have been defined and the initial plan for configurability has been established, the needed information for configuration matrices starts to appear. The initial plan for configurability can be considered to be the basis for configuration matrices as variety, commonality, differentiation, outsourcing and upgradeability are considered. This plan will guide the product development through the entire product development process. After the functional decomposition of the product and the first stage of MBI there is a possibility to locate the initial modules into the matrix representation. Although this is possible, the MBI method should be done first and after the modular structures of the product have been established the configuration matrices can be set up (Figure 24).

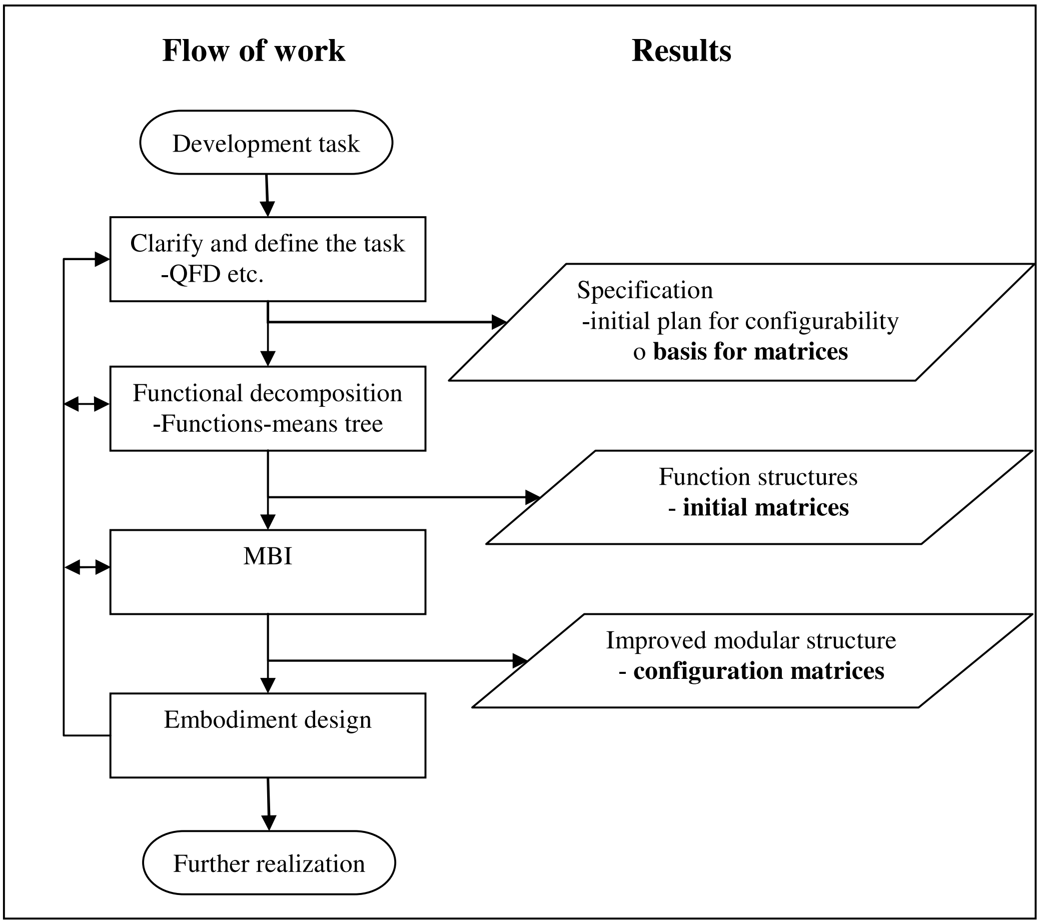


Figure 24. The flow of work in MBI (Aarnio 2003) connected to the formation of matrices

As mentioned before, the configuration matrices are formed by considering the marketing options and the modules simultaneously. The modules and the product architectures should be designed to holistically meet the needs of the initial plan for configurability. As Bongulielmi (2003) and Sekolec (2003) present, the K- and V-matrix representation can be used during the design process to consider internal and external variety of the product structure. As the configuration matrices presented by this study are very close to the ideas of K- and V-matrices, the features of the matrix representations are also similar. Configuration matrices presented in this study are actually a subset of matrices presented by Bongulielmi (2002 and 2003). Next to this, the K- and V-matrices could also be seen as subset of configuration matrices presented in this thesis when considering that also other issues than only the configuration task is considered. Thus, configuration matrices also include the platform to provide the generic product structure to be used in multiple operations. The concept of producibility forces the use of a wider view that considers product structures holistically.

Considering Figure 24, the MBI method could be changed to be any of the presented methods used to modularize the product. Since nearly all the product structuring methods use functional decomposition of the product structure prior to the modularization efforts, the process shown in Figure 24 can be considered general in many sense. If the company sees that configurable products are essential for the business, there should be a company-wide understanding of the impacts the configurable products have on operations. If the configurability and the configuration knowledge are not generated by the design department, iterations between different functions are more than likely. This is the reason why the configuration matrices should be established in an early phase of the design process when also considering prototyping and ramp-up as sub processes for the entire product design process.

The matrices can be created after modularization of the product structure has been established and there is a clear understanding and consensus of what the saleable options of the product will be. If the functional decomposition and the following integration of modules are defined considering configurability of the product structure, the matrix representation can be established by the following process:

- define the generic list of options
- define the generic product structure
- define the variable part of the product structure
- define the base machine part of the product structure
- define the sequence of the saleable options
- generate the square matrix by listing the options and modules according to the sequence
- define the dependencies of the modules with respect to the options
- finalize the sequence of the matrix

The generic list of options can be drawn from the initial plan for configurability and the generic product structure from the module listing of the product decomposition and the following integration. The part of the process where the base machine and the variable proportion is defined can be seen as revealing at least the inner platform of the product family. The base machine proportion is considered to be standard in all the configurations that can be drawn from the modular system. This part is of great importance since it can be modularized by any means (Lehtonen et al. 2003). This is because it is standard, i.e. this is the proportion of the product structure that can be produced most cost-effectively and designed easiest considering all the possible stakeholders. It needs to be addressed though that the type of modularity affects the usability of the matrix representation very much. This is the reason why the evolution model presented by Lehtonen et al. (2003) is followed in order to produce the wanted results. The rest of the matrix formulation is iterative and the sequence of the matrix representation is totally dependent on the experience of the designer building the matrices. It is also noteworthy that there is no database introduced to handle the matrices, but the maintenance and building should be done by others means in order to secure the visibility of the product structure during the design phase.

If the matrix representation is used during the product design phase, the part of preplanning is very essential since the customer needs have to be established and the product structure decomposed according to the specification drawn from the customer needs. This is not to say that changes cannot happen during the design phase but to understand the importance of preplanning or product planning presented by Ulrich and Eppinger (2000). Actually the configuration matrices can be used to effectively consider the changes during the design phase by analyzing the dependencies of the matrices. The problematic area is that the final matrices show the dependencies between options and modules, i.e. there is no strict presentation of the module interfaces available. The emphasis should be targeted on the base machine versus the variable part of the machine since, as mentioned before, there is cost related to variety, i.e. the bigger the standard part, the more cost-effective the structure is. The idea is that the product structure follows a clear defined principle (Lehtonen et al. 2003) and the modules are designed to fulfill the requirements of different stakeholders. This means that also the varied part of the product structure should be designed considering the needs of the stakeholders, i.e. the modules should be designed to meet the needs of the customer as well as the needs of the manufacturing system in a broad sense. This also implies that there is a possibility that module interfaces need to be redesigned according to new requirements of the stakeholders, i.e. the standard part of the structure gets smaller as the varied part of the product structure grows bigger in size.

Considering the generic process presented by Ulrich and Eppinger (2000), the importance of prototypes and production ramp up is evident. Configurable products also put pressure to the marketing side since the configurator, both manual and automatic, needs to be set up before the production ramp-up at the latest. There is a need to establish very systematic ways to handle configurable products from product design to production in order to secure short lead times.

3.4.2. Process for generating configuration matrices for existing products

When considering existing products the need for well established configuration knowledge is as important as the establishment of configuration knowledge during the product development process. It has been reported (for example, Forza and Salvador 2002a) that modeling configuration models can be difficult and time consuming and usually only few people in the company know how to configure a specific product. This has also been the case during this research, i.e. there is a need to visualize the configuration knowledge for existing products.

As the configuration knowledge is usually cumulated to the production department it is an ideal place to start collecting the needed material for the establishment of the configuration knowledge. The process for modeling existing products is as follows:

- gather the required configuration knowledge by interviewing experts
- define the generic list of options
- define the generic product structure
- define the variable part of the product structure
- define the base machine part of the product structure
- define the sequence of the saleable options
- generate the square matrix by listing the options and modules according to the sequence
- define the dependencies of the modules with respect to the options
- finalize the sequence of the matrix

A major part of the process is similar to the previously presented process used during the product development. There are still differences related to the information gathering during the establishment of the configuration knowledge. All the knowledge related to configuration is gathered in the first phase. There is usually some way of presenting the knowledge in addition to the hidden knowledge stored in the experts' minds. The task is to reveal all this knowledge by interviews. The problem is that the combinations of the options as well as complex company specific issues disturb this phase while the inconsistencies between production and product design in the context of product structure can be very tedious to solve. The main thing is to keep the matrix representation in mind and use an experienced matrix builder in order to succeed. For gathering purposes the following chart can be used, Figure 25.

Base machine - varied part of the structure

Base machine modules

M00001 HYDRAULICS
M00002 TOOLS PACKAGE
M00003 CLUTCH
M00004 ENGINE
M00005 CABELS
M00006 HYDRAUL PUMP

Options and their module selections — **Note**

IF ROTATOR XXX SELECTED INCLUDE,
M00007 ROTATOR
IF ROTATOR XXY SELECTED INCLUDE,
M00008 ROTATOR

Note: IF ROTATOR XXX SELECTED, CRANE XXY CANNOT BE SELECTED

IF CRANE XXX SELECTED INCLUDE,
M00009 CRANE
IF CRANE XXY SELECTED INCLUDE,
M00010 CRANE

IF ROTATOR XXX AND CRANE XXX SELECTED INCLUDE
M00011 CRANE HYDRAULICS
IF ROTATOR XXY AND CRANE XXY SELECTED INCLUDE
M00012 CRANE HYDRAULICS
IF ROTATOR XXY AND CRANE XXX SELECTED INCLUDE
M00013 CRANE HYDRAULICS

IF LIGHTS SELECTED INCLUDE
M00014 LIGHTS
M00015 LIGHTS
IF LIGHTS AND ROTATOR XXX AND CRANE XXY SELECTED INCLUDE
M00016 LIGHTS

Figure 25. Example chart for gathering configuration knowledge

After all the possibilities and modules have been defined, the chart shown in Figure 25 is used to define the configuration matrix. As shown in Figure 25, the base machine and the variable part of the product have been defined and the preliminary work for configuration sequence has also been defined since all the dependencies must be declared at this point. The inconsistencies of the product structure related to configurability can be easily noticed since the tediousness of the work grows as the complexity of the modularity of the product increases. The next phase is to define the configuration matrix for this specific data gathered. The first step for generating the matrices is to plot the generic list of features and their options into the matrix followed by the related modules. After this the chart shown in Figure 25 is used to plot the dependencies into the matrix representation. Figure 26 shows the configuration matrix derived from the above presented chart (Figure 25).

Model Config	ROTATOR XXX	ROTATOR XXY	M00007 ROTATOR	M00008 ROTATOR	CRANE XXX	CRANE XXY	M00009 CRANE	M00010 CRANE	M00011 CRANE HYDRAULICS	M00012 CRANE HYDRAULICS	M00013 CRANE HYDRAULICS	LIGHTS	NO LIGHTS	M00014 LIGHTS	M00015 LIGHTS	M00016 LIGHTS	BASE MACHINE	M00001 HYDRAULICS	M00002 TOOLS PACKAGE	M00003 CLUTCH	M00004 ENGINE	M00005 CABELS	M00006 HYDRAULIC PUMP
ROTATOR XXX	o	x	x		x		x	x	x					x	x	x							
ROTATOR XXY	x	o		x	x	x	x	x		x	x			x	x								
M00007 ROTATOR	x		o																				
M00008 ROTATOR		x		o																			
CRANE XXX					o	x	x		x		x			x	x								
CRANE XXY					x	o		x		x				x	x	x							
M00009 CRANE					x		o																
M00010 CRANE						x		o															
M00011 CRANE HYDRAULICS					x				o														
M00012 CRANE HYDRAULICS						x				o													
M00013 CRANE HYDRAULICS					x						o												
LIGHTS												o	x	x	x	x							
NO LIGHTS												x	o										
M00014 LIGHTS												x		o	x	x							
M00015 LIGHTS												x		x	o	x							
M00016 LIGHTS												x		x	x	o							
BASE MACHINE																	o						
M00001 HYDRAULICS																		o					
M00002 TOOLS PACKAGE																			o				
M00003 CLUTCH																				o			
M00004 ENGINE																					o		
M00005 CABELS																						o	
M00006 HYDRAULIC PUMP																							o

Figure 26. Configuration matrix derived from the example shown in Figure 25

As shown in Figure 26, all the possible marketing configuration rules are presented in the crossing sections of options and production configuration rules can be found from the gray areas as the modules are linked into the options. Considering the rotator selection, there is one marketing rule that forbids the selection of crane xxy if rotator xxx is selected. Considering this example there is no pre-designed crane hydraulics available for the combination of rotator xxx and crane xxy, i.e. the possibility to choose wrong combinations of options is eliminated. The matrix shown in Figure 26 will be used in section 3.8 to provide examples of the configurator developed during this research.

As the example in Figure 26 is very limited in size, there is no need to rearrange the sequence of the configuration task since it is very clear that there is no other possibility. When the matrices grow bigger in size, there is a tendency that the first features and their options can have dependencies between many other options linking the structure together in different ways. When the matrix is ready as shown in Figure 26, the applications developed can be used to analyze the matrix and the product is ready to be configured by manually using the matrix representation or automatically by using configurators.

3.5. The role of product structure and configuration matrices

Product structure can be seen as an integrative element between different operations of the company, i.e. between design, production, marketing and sales, purchasing (included in the production in the following text) and after sales. As modularity is introduced into the product structures all the stakeholders mentioned should be considered to avoid sub-optimization. The role of product structure in organization is presented in Figure 27.

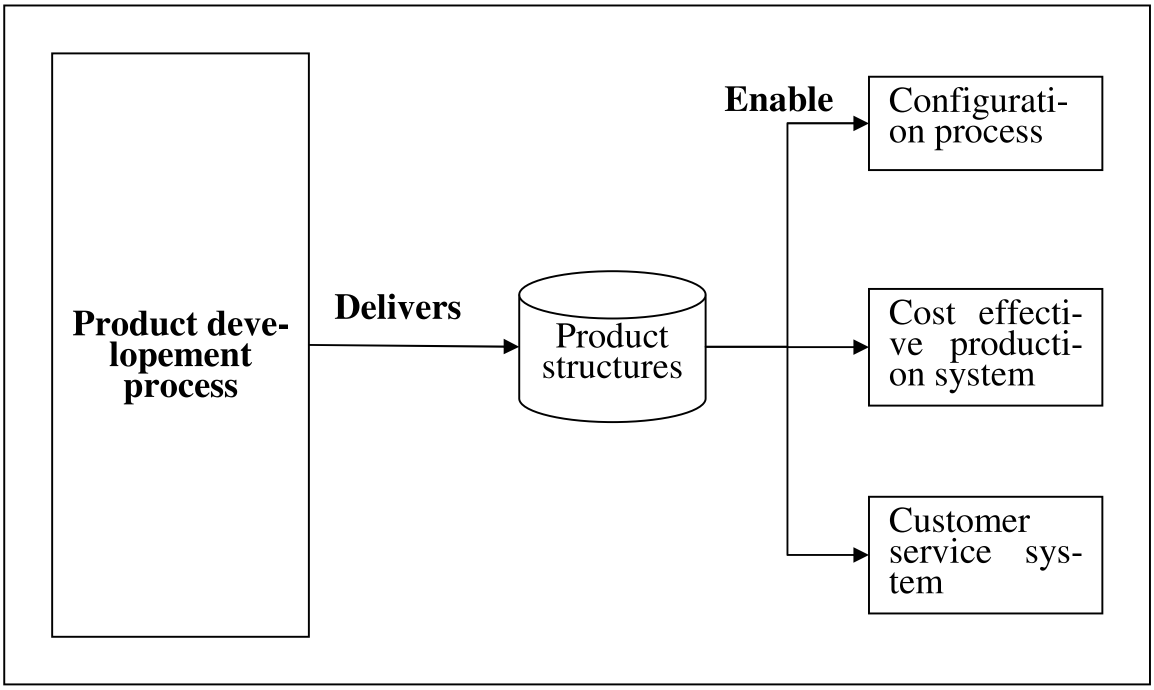


Figure 27. The role of product structure in organization

Product development can be seen as a process that delivers the product structure to be used in marketing and sales (configuration process), in after sales (customer service system) and in production (production system). As product development produces product structures, it can be considered as the main process whereas the rest of the processes mainly take advantage of the produced structures and give constraints and requirements to the product development concerning the product structure. In the world of mass customization (Pine 1993) the above Figure 27 can be thought to possess the following properties:

- An effective product order process for configuration (marketing and sales)
- An effective customer service process (after sales)
- A cost-effective production process (production)

Thus, there is a need to have a meaningfully decomposed product structure that allows the use of an automatic configuration process, which is followed by a cost-effective production to produce customer specific product individuals, which is followed by a rapid after sales to serve customers. The product structure produced by the product development process affects all the other processes, i.e. the product structure cannot be sub-optimized to serve only one stakeholder efficiently.

Peltonen et al. (1998) present the idea that the configurations of a product should be concrete, valid and complete, i.e. the elements of the configuration models ask all the necessary questions, the complete configuration answers all the asked questions, and there are no wrong answers to make valid configurations. The main goal for the configuration matrices is to satisfy the needs of the configuration model to be concrete, valid and complete in order to generate concrete, valid and complete configurations of a product individual. Considering the organization, product configuration and configuration matrices bring constraints and some rules in order to cope with the configurable products and even further to cope with the automatic configuration process. Thus, the configuration process alters the surroundings of the company in the context of mass customization. While configurable products are developed, the visibility of the configuration rules needs to be addressed in order to cope with the environment developed by the selected competitive strategy. Configuration models and their maintenance are critical in all situations while executing build-to-order strategies. Configuration models and their maintenance are required to retain the knowledge related to the products and enable the use of an effective configuration process. Mesihovic and Malmqvist (2004) are concentrating on developing processes for maintaining configuration models and they have established a generic process for this application. The field of configuration matrices is to cope with the disturbances that affect the sales delivery process by visualizing the configuration knowledge, i.e. the change processes of marketing and design cause disturbance related to configuration models and in order to effectively cope with the environment, well-defined configuration models are needed. In Figure 28 the role of the matrices in the organizational context is presented.

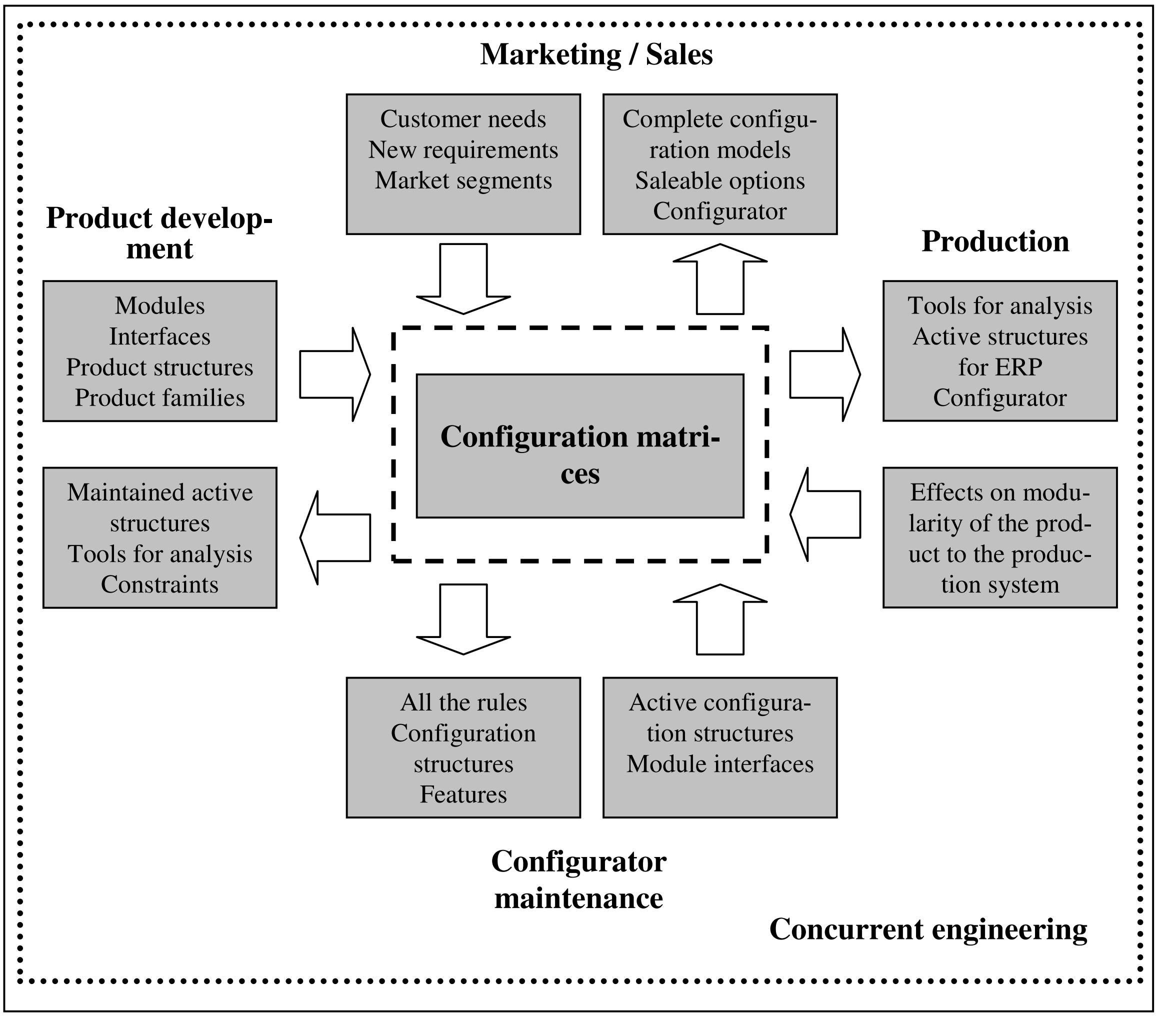


Figure 28. Role of the configuration matrices

The configuration task, both automatic and manual, uses the knowledge provided by the configuration matrices. The configurator maintenance takes care of the lifetime prolongation of the configurator and also of the maintenance of the matrices. It needs to be noted here that the amount of configurator maintenance depends heavily on the type of configurator and configuration models. Concurrent engineering is defined by Parkinson et al. (1996) as follows:

- "The consideration of all downstream activities which are likely to affect the product's life cycle at the products design stage"

Parkinson et al. (1996) continue:

- "In respect of product design, the designer or design team should be made aware at all stages of any implications that decisions taken at this stage have upon the final manufacturing specification and its resulting outcome."

The definition given by Parkinson (1996) provides the idea that all the stakeholders (applicable) should be present when considering decisions critical to other operations. Concurrent engineering in the context of configuration matrices clarifies these issues. The inputs (Figure 28), which the configurator matrices need, are regarded to be part of the product design process, i.e. the part of concurrent engineering in the context of generating matrices and developing product structures that satisfy the needs of many stakeholders of the company. Concurrent engineering as defined by Parkinson et al. (1996) consider the entire lifecycle of the product, i.e. the configuration knowledge is present to support many activities during the product's life cycle. Also the implications that certain kinds of product structures can have on operations are critical to understand at an early stage during the product development process. As seen in Figure 28, the outputs that the configuration matrices give are considered to be a part of the routine tasks involving all the main processes in organization.

The part of concurrent engineering is highlighted since when designing a product, the types of modules and modularization, it is critical to consider the entire company and its processes. If the configurability is the main driver (Aarnio 2003), the second biggest should be the inputs from manufacturing. If the aspect of manufacturing is involved in the context of modular product structures, the possibility of a producible product increases. The producible product in the context of matrix representation means that an easy configurability and the systematic approach to the similarity between the production system and product structure is secured. Configuration matrices also help to decide product platforms which will be discussed later in this chapter. The ability of the company to provide this type of modularity implicates good modularization capabilities from product design processes.

3.6. The evolution of modularity and configuration matrices

Lehtonen et al. (2003) present the evolution model of modularity in Figure 14. This presentation of types of modularity gives an insight into the practical use of modularity and its benefits for business strategies. During this research it became very clear that the level of understanding the different types of modularity holistically and company-wide is critical when changing the routines for product development in issues relating to modularity. Considering only the product, production, marketing or product development, the decision of the type of modularity to choose is easier than in the case of holistically defining modularity into the product structures.

From the experiences gathered, the evolution model presented by Lehtonen et al. (2003) was very closely tied to the evolution of modularity in the case company, as will be presented in chapter 4. If the evolution model is followed very closely, the shift from one level to another is more easily handled than skipping over one stage of evolution. For configurable products functional modularity is the most efficient. Lehtonen et al. (2003) consider that functional modularity of the product support marketing and sales, product design and configuration, i.e. it is best for configurable products. The problem in functional structures is the loss of similarity between production and product structure, which in turn is at its best in structural (assembly based) modularity one level below functional modularity. The main difference between these two levels is that configurability is much more complex in structural modularity than in functional modularity. Also the maintenance of drawings and designs is much more complicated in the structural type of modularity. The shift to the third level is discussed in detail in chapter 4 while considering the case and the use of configuration matrices while shifting from functional modularity into the customer oriented platform based modularity. According to Lehtonen et al. (2003), the idea for the third level is to decrease the cost of customer variation by restructuring the product, i.e. the point is to enable more standard structures compared to functional modularity. The lower part of stage three in Figure 14 presents the standard part of the product structure which provides the cost-effectiveness for configurable products. This can now be seen as shifting back to the structural modularity in a sense that the standard part of the product structure can be modularized by structural modularity. This is easily defined by the configuration matrices, i.e. the base machine presented above equals the standard proportion. Even if the last stage provides the highest level in Figure 14, it can be argued that it is related to the company specific capability of designing modularity in order to provide enough structural modularization to provide cost efficiency. Thus, it can be stated that when the company has enough capability to provide modularity, it can turn back to structural modularity to provide configurability and cost efficiency. Finally, the future is the shift to dynamic modularization and concentrating on the whole life cycle of a product. Dynamic modularization is based on a dynamic platform and ideas of mass customization (Riitahuhta, 2000).

The power of configuration matrices, considering the evolution of modularity, is their ability to provide insights into the existing type of modularity possessed by a product structure. If the knowledge of the generic product structure is dispersed and not systematically documented, the level of modularity can be hard to estimate or understand. While it is hard to holistically understand modularity in these cases, it is also impossible to setup a configurator without documented configuration knowledge.

3.7. Configuration process and matrices in changing environment

From the experiences gathered in the case study of this thesis, there are various ways to execute product configuration. First of all, there is the manual configuration of products. This type of configuration is usually the most flexible in nature and it can tolerate most effectively disturbances from other stakeholders concerning the configuration process. If the manual configuration is under control, the product structures and change management support configuration, there are possibilities to create an automated configuration process. For this approach there can be found many possibilities as the number of configuration software suppliers grows. It is possible to configure in ERP systems, in a PDM system or in between by using a third party supplied configurator. This is obviously a company specific decision and depends on the software strategies defined for the company. These software considerations usually involve the role of the configurator as well as the need for long term management of the configuration models as well as configured product configurations. The final possibility is to develop an in-house configurator to match the configurator and the company specific processes related to the task of configuration.

The configuration process alters between companies as the products and organizations as well as processes differ. The main aspect shared in all companies executing a configuration process (manual or automated) is the management of the configuration models. Considering the entire life cycle of the product the changes can be generated from at least the following sources (Tiihonen and Soininen 1997):

- Customer requirements change
- New products are developed
- New functions and possibilities are added to existing products
- Functions and possibilities are removed as they become obsolete

Configurability is not only a product issue, but in order for the configuration process to work properly the design processes, organization as well as marketing and sales organization and processes need to be re-engineered accordingly. For the environment of the configurable products and configuration matrices the problem is the changes affecting the configuration model, i.e. configuration matrices. In this research, two main sources of disturbance were defined: changes from product design and changes from marketing and sales. As manual configuration using matrices for product configuration algorithm is considered, it can allow lots of disturbance from the above-defined sources. The problems usually occur in manufacturing and the penalty is paid in growth of inventory levels and possible late deliveries of parts, components and even products.

While the automatic configuration process is considered, problems occur in the maintenance of the configuration models, i.e. the databases of the configurator need to be frequently changed. The change management presented by Mesihovic and Malmqvist (2004) can be used for systematizing change processes, but at the same time the frequency of the change proposals should be analyzed in order to estimate the rate of change concerning the configuration models.

Configuration matrices have been developed to reveal the configuration knowledge and they are especially used in a situation where the maintainability of the configuration models is hard due to the frequent changes. Even if the configuration matrices are used, the problems considering the organization are still related to the frequency of the changes reflecting to the configuration models. The environment of the configuration process is shown in Figure 29. The changes defined above by Tiihonen and Soininen are included in Figure 29 together with the changes defined during the case study of this thesis.

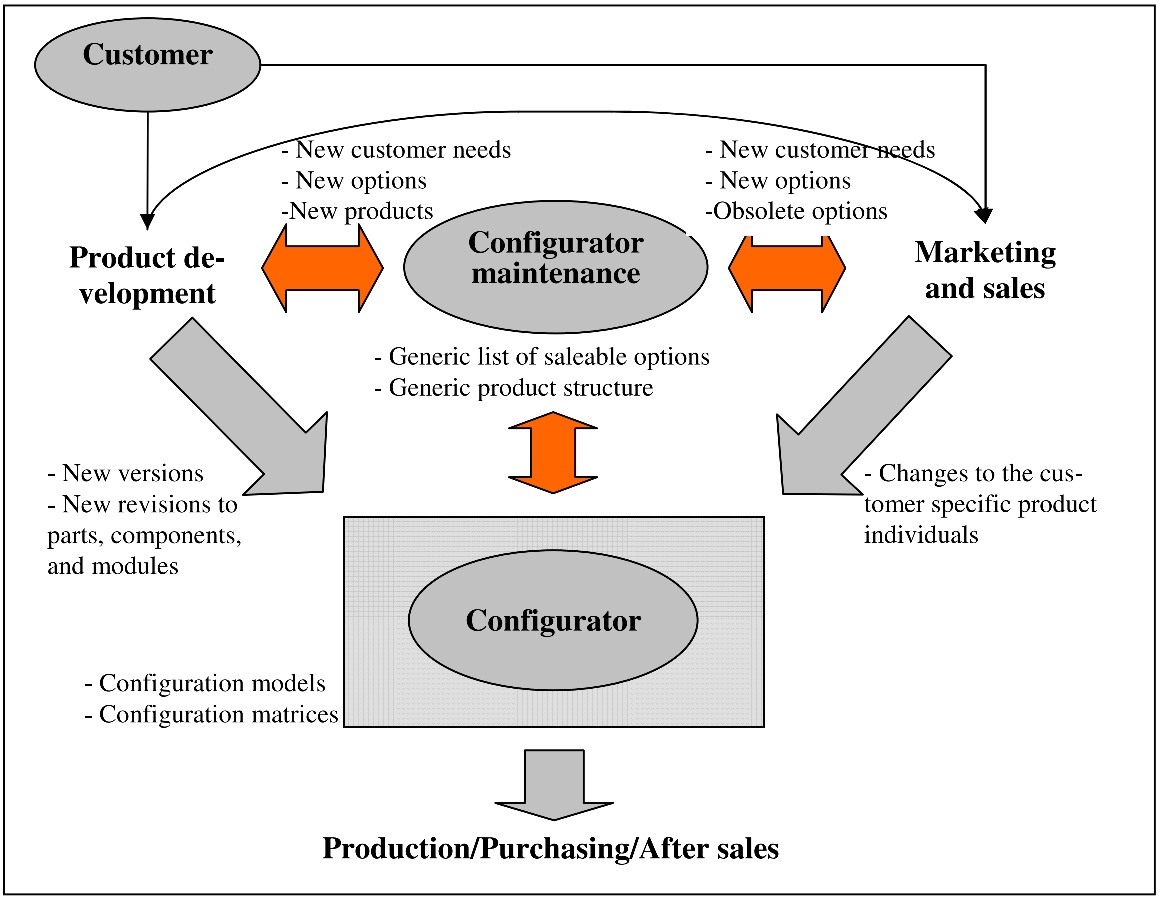


Figure 29. The environment of product configuration

The definition of revision in this thesis is:

- Parts and components can be used after revision change simultaneously with the old components and parts, i.e. no change scheduling is needed for revision change

The definition of version is:
- Parts, components and modules that are not compatible with old versions, i.e. scheduling of version change is required

Considering this situation the problem is that new versions of modules affect the configuration model, i.e. the configuration matrices and the configuration model in a possible configurator is affected. This now implies that there is a need for versioning matrices and configuration models in order to satisfy the needs of the life cycle management, i.e. there is a need to secure the requirements from after sales to define the configuration model used at a time and even reconfigure a specific product individual. Even if new versions do not affect the module interfaces, i.e. the configuration matrices are not affected, the problems start to occur, for example, in purchasing and pricing as new revision inflate the existing pricing system. This is discussed in detail in section 3.8.

In Figure 29, the gray arrows represent the operational work to the existing products and the red arrows are related to new products and life cycle issues of existing products. While the changes from product development can be considered to be improvements on the parts, components and modules of the product, the marketing changes are usually concentrating on changing the existing, once configured product individual (gray arrows in Figure 29). The red arrows represent new product development as well as the development of new functions and removal of old ones. These changes alter the generic product structure and also the generic list of options to be used during configuring the product. Marketing and product design are heavily integrated considering the customer needs. The workload of configuration maintenance is related to the rate and to the nature of the changes. If the options of the product are altered frequently, the work of configurator maintenance can be tedious as well as if the interfaces of the modules change causing the problems in the combinations of the options related to the selection of a certain module. The easiest changes are the revision changes as the interfaces of the modules and the options structure are not affected.

Even if the product structure is already configured, customers sometimes need to change the options selected, i.e. the product individual is changed during the order delivery process. The point where these types of changes can take place is critical since many times changes of this type have tremendous effects on production.

New options and features are likely to appear from marketing and product design as the life cycle of the product progresses. These changes can be related to new customer needs or advances in technology. Also a poor preliminary design affects the already launched products in the form of new options and features to be added after the product launch. Also the revision related changes affect product configuration since the point when the configuration is established determines the product structure, i.e. the revisions are locked to the product individual. This is a matter of product life cycle management as after sales use the product structure configured during the order delivery process.

Even if the configuration matrices can be integrated to support the configurator, the problem is that continuous changes during the life cycle of the configuration model cannot be allowed. This is mainly due to the fact that if the processes of product design and marketing allow short lead times by generating incomplete designs or specifications, there is no sense in trying to automate the configuration process. This would lead to an enormous amount of configuration model versions during the life cycle of the product.

It is easy to conclude that if a configurable product is the first step to reach mass customization, the second is to automate the configuration process. Either of these steps cannot be accomplished without redesigning processes related to change management, product design, marketing and production. To even reach the point where the generic change process presented by Mesihovic and Malmqvist (2004) can be used, a lot of work needs to be done especially in product design and marketing for types of companies considered during this research. The sources of changes related to configuration models presented above are all causing disturbance to the configuration process through configuration models. The reason to visualize the configuration knowledge is important in order to handle the changing environment of the configurator. The rate of change can be thought to arise from the markets and from the processes inside the company, i.e. the changes are inevitable but processes can be designed to minimize the rate of change from inside the company. As the configuration models control the configuration process, the issues related to configurable products affect all the company functions, i.e. product design, production, marketing and after sales are affected. In order to support the processes in a changing environment, section 3.8 presents a configurator that is robust against change situations.

3.8. Configuration matrices as basis for a configurator

The properties of configuration matrices have been considered in the above discussion. During this research the main objective was not to build a configurator, but to use an ERP, PDM or third party supplied configurator, and to use configurator matrices to automatically provide the needed information to maintain the configurator. The main idea behind the configurator established in this thesis is to integrate the following factors into the configurator as automatic features:

- Feature-based pricing
- Feature-based pricing maintenance
- Production configuration rules maintenance
- Marketing configuration rules maintenance
- Algorithms to decide the validity of the configuration rules

The above features are the most critical parts for the configurator while simultaneously they provide new features that conventional configurators do not. The conventional configurators usually concentrate on providing an effective rules generator that the company can use to maintain the configurator rules. This is one of the main issues that provide stiffness and non-value added work to the system since all the rules need to be manually entered and while changes to the products are imperative in configurable products during their lifecycle, the risk of an invalid configuration model increases (see section 3.7). This also implies that the workload of configurator maintenance increases and in order to keep the configurator databases valid there needs to be a way to present the configuration models next to the databases of the configurator. The comparison between different configurators and the properties related to configurator matrices lead to a conclusion that a self-made configurator would be the most beneficial solution (see appendix 4).

The main idea for the self-made configurator is to provide a flexible configurator with a possibility to minimize the work related to configurator maintenance. Figure 30 establishes the grounds for an automatic configurator provided by the configurator matrices.

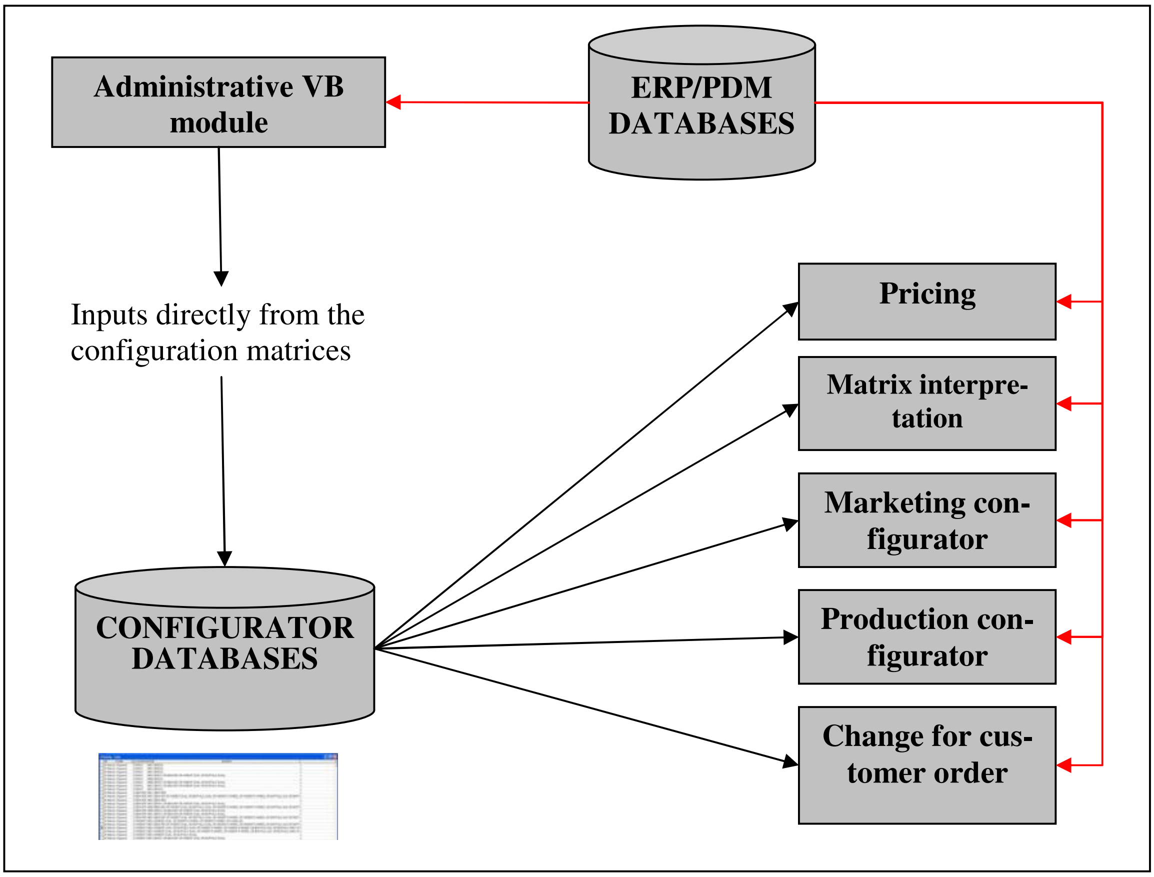


Figure 30. The concept for configuration matrix-based configurator

As shown in Figure 30, the main idea is that there is a VB-module that provides the configuration related knowledge by integrating the company specific ERP/PDM databases with the configuration matrices and establishes automatically the databases needed. These databases are then used by various programs including marketing and production configurators and pricing applications. Also these applications use ERP/PDM databases if necessary. As the databases are critical to the entire concept next to the listed integrated factors presented above, they are all considered separately in the following sections.

3.8.1. Databases for matrix-based configurator

The databases for this matrix-based configurator provide all the needed information when establishing a configuration for a customer specific order. The databases are established automatically from the configuration matrices stored in Excel. The administrative VB module presented in Figure 30 translates the knowledge from configuration matrices into the databases. Figure 31 presents the interface.

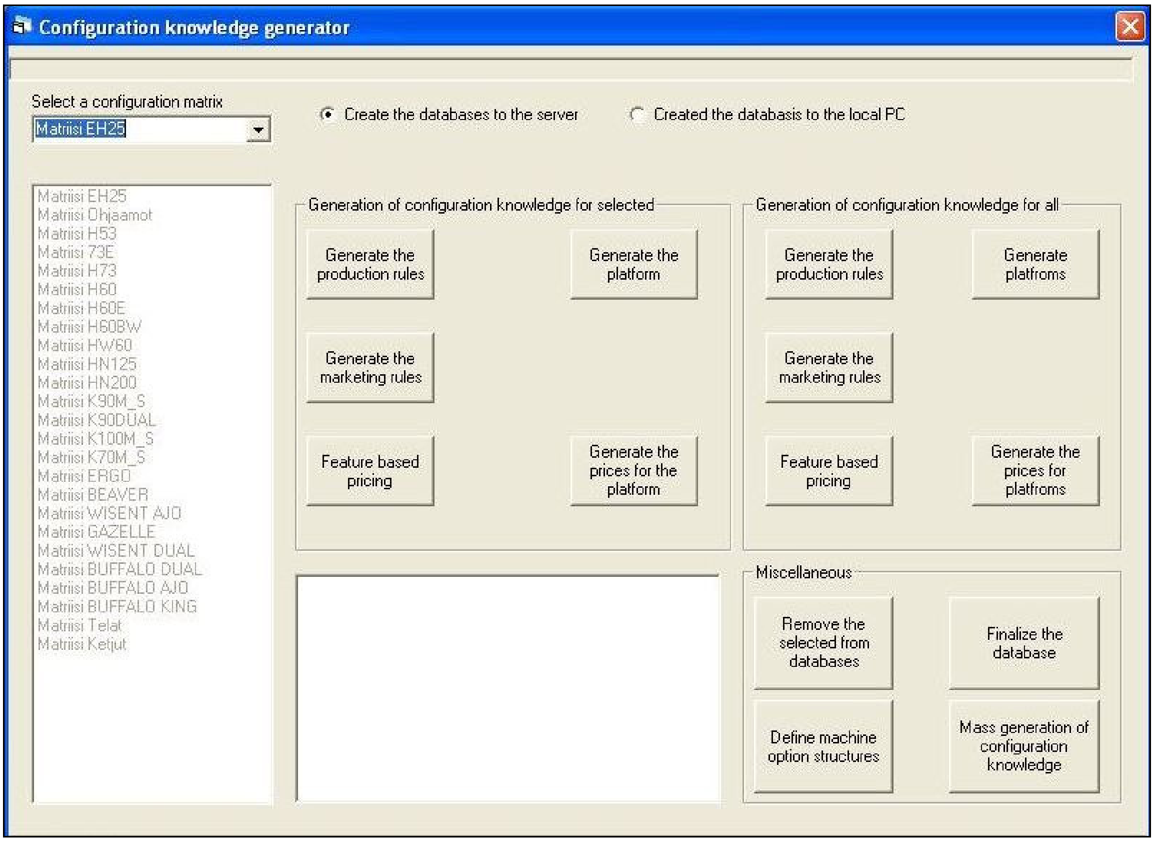


Figure 31. The configuration knowledge generator

All the modular systems, i.e. configuration matrices are presented in the list box on the left hand side in Figure 31. The possibilities to generate configuration knowledge are to generate the knowledge into the server or into the local PC. There are also possibilities to generate the knowledge only from one matrix, all the matrices or to generate the knowledge through mass generation. Mass generation converts all the configuration matrices into databases that can be then used for configuration purposes. It is noteworthy that this creation of configuration knowledge is automatically done, i.e. the configurator is ready to be used right after the mass generation is done. The procedure that takes place whenever the mass generation of configuration knowledge is selected, is as follows:

- Generate all the rules for production configurator for all the modular systems
- Generate all the rules for marketing configurator for all the modular systems
- Generate all the platform structures for all the modular systems
- Generate all the feature costs for all the modular systems
- Generate all the costs for platform structures for all the modular systems
- Finalize the databases

The way the generation of configuration rules is done is through different algorithms that study the matrices. For production configuration rules the algorithm studies the configuration matrices vertically to decide the configuration rules. The database includes strings such as (ROTATOR XXX OR ROTATOR XXY) AND (CRANE XXX OR CRANE XXY) AND (LIGHTS). This rule implies that the module related to this rule will be selected when there has been a selection of one of the rotators, one of the cranes and lights. The similar rule is present in configuration matrices when studying the matrix manually, see Figure 26.

For marketing rules, the algorithm studies the configuration matrices horizontally to decide if there are dependencies between the options to be selected. The algorithm analyzes the configuration matrices by deciding not valid and valid options whenever there is a rule related to an option. The algorithm studies all the rows and columns of the configuration matrices, but only those that have rules are established into the database. These rules are used for the marketing configurator to make options available or unavailable.

Platform structures in configuration matrices present the modules that will always be included into the configured product. These modules will be defined after all the configuration rules have been established. The reason why these modules are also defined is because by doing this the entire configured product structure can be established, i.e. the variable proportion next to the standard part of the structure is defined.

Feature-based pricing is further discussed in 3.8.2. Finally, the databases are finalized. This process makes the configuration databases ready to be used for the configurator. The main task for this last phase is to check any inconsistencies in the databases and to define the possible double rules for configuration models that can be present. After all the above steps the databases are defined and the configurator is ready to be used. Next, the feature based pricing is discussed in depth since it is one of the cornerstones for the configurator.

3.8.2. Feature based pricing

Almost all the conventional configurators provide means to include pricing information when configuring the customer order. The problem is that this type of information can be very tough to define as the module combinations selected can be combinations of many options selected. If this is the case, the problem is that defining these combinations manually becomes very time consuming and frustrating. The ideal situation would be that one option selection would not make combinations with other options, i.e. the complex world of module selections through option combinations would be eliminated. This is not always the case though.

As presented in Figure 31, there is a procedure to define feature-based costs. This is again a property of the configuration matrix, which makes it possible to define all the module combinations that can be retrieved from the option structure and their combinations. For example, there might be three different features that affect the module selection. If all of these features have three options, there will be total of 27 (3 x 3 x 3) possible combinations. This also means that when all the needed information is included in configuration matrices there is no possibility to get a zero cost feature, i.e. all the possibilities will have their own costs. The algorithm for this procedure is fairly complex since it:

- Generates autonomously all the possible configuration rules

- Configures according to the generated configuration rules

The first situation implies that the algorithm decides for the above example (27 combinations) what the possible 27 strings that equal configuration rules are. Before this can be done, the algorithm defines all the possible features that are involved when considering an option selection. Thus, when one option is selected, the algorithm defines all the features and their options that make combinations with the selected option. After this definition the application can define the possible configuration rules. Using these formulated configuration rules the built-in configurator will configure all the possible outcomes of the entire feature. For the example used in this section there will be 27 different module combinations defined with their respective costs. For this application the integration between ERP databases needs to be in place in order to be able to define the inventory value for each part and component. After each configuration the application decides the latest module revision, defines the BOM for this revision, and finally adds up all the costs retrieved from the ERP system.

For the configurator knowledge generator the costs will be established automatically without showing each and every module combination that resulted from the configuration tasks. There is also an ASP application that shows the results for every string that has been configured during the pricing sequence. The way this ASP application works is that the wanted option is selected and the application then uses the information from the configuration databases as shown in Figure 30 to come up with a solution presented in Figure 32. The feature selected in Figure 32 has dependencies with two other features. The first feature has two options as well as the second one which makes a total of 4 different combinations to be defined through configuring the alternatives (see Figure 26 for the matrix representation). In Figure 32 the first two possibilities are presented. To clarify, the root option for this example is “LIGHTS” while the options of the features that the root option is dependent on are as follows:

- Feature 1
    - ROTATOR XXX
    - ROTATOR XXY
- Feature 2
    - CRANE XXX
    - CRANE XXY

As shown in Figure 32 the first module combination is decided by option string “LIGHTS;ROTATOR XXX;CRANE XXX”. This string will provide modules M00014 and M00015 with a cost of 3962.89 [currency units]. The next box presents the following situation where the option string is “LIGHTS;ROTATOR XXX;CRANE XXY”. This string provides modules M00014, M00015 and M00016 with a cost of 6080.10 [currency units].

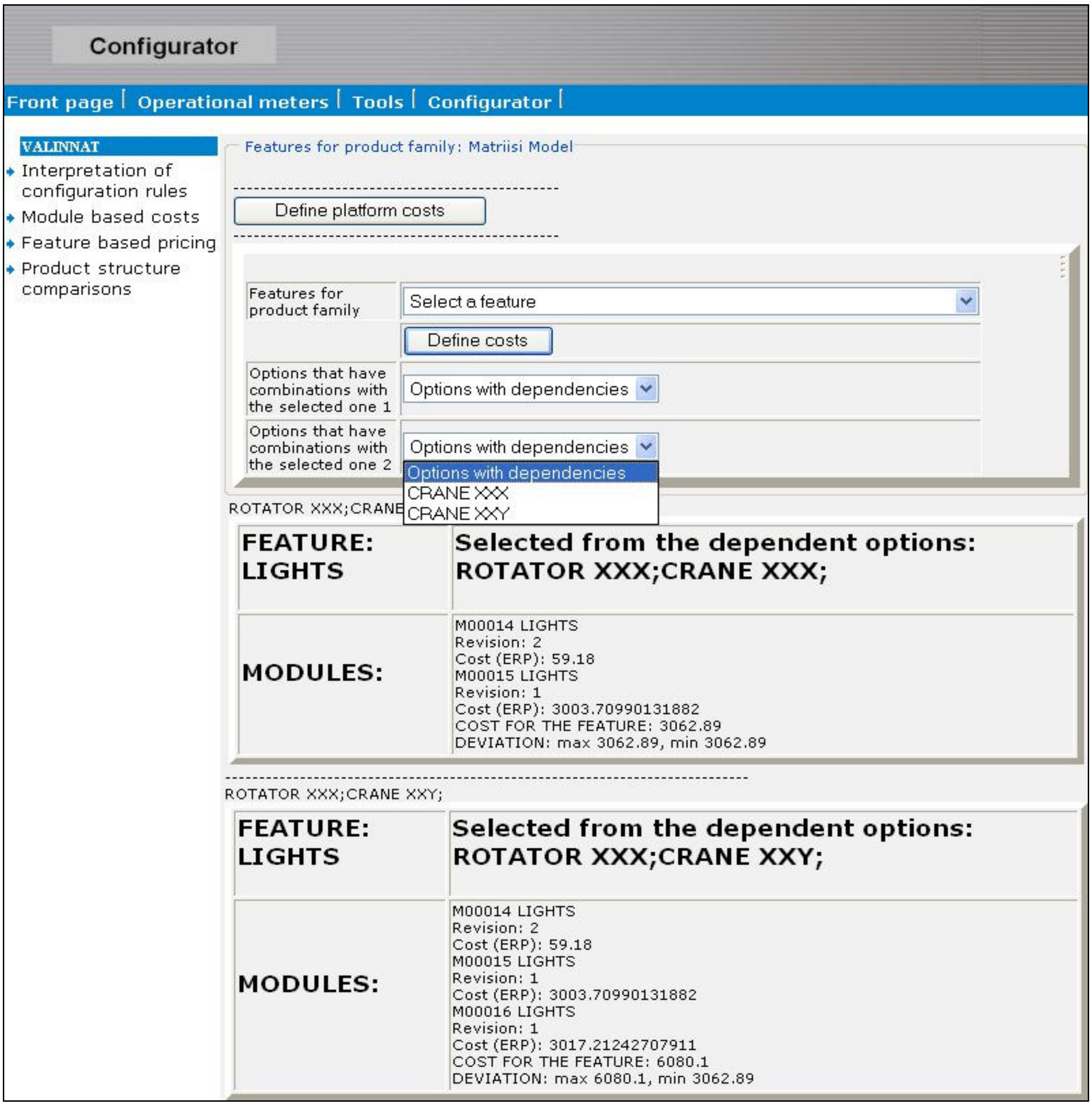
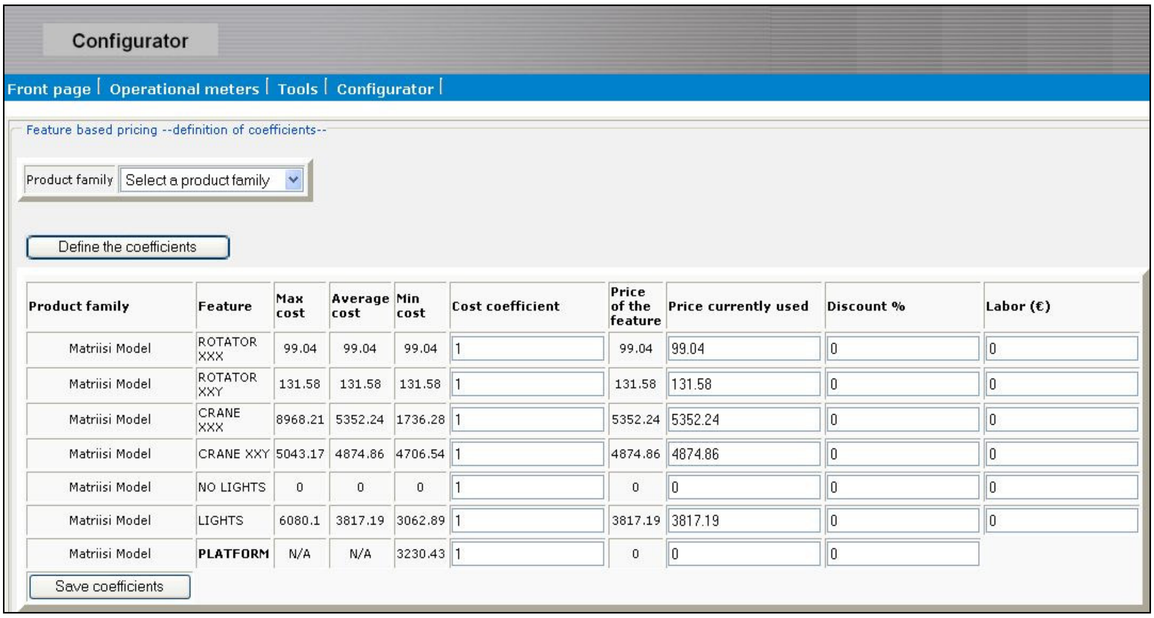


Figure 32. Defining costs for an option with dependencies

By using this approach all the four possible option string combinations are considered and the costs estimated. Also the deviation in the costs is recorded and the result from the application is the mean value of the costs with the minimum and the maximum values.

When the database generator is considered, the procedure presented above will iterate through every modular system, i.e. all the options from all the matrices are treated the same way. This will result in a database with all the saleable options, their maximum, average and minimum values of costs. The interface presented in Figure 33 will provide insights how the cost data can be used to determine option prices. Figure 33 shows the results for the configuration model used also in Figure 32 with its costs and selling prices after the configuration rules generator has established the costs for the model.



Product family	Feature	Max cost	Average cost	Min cost	Cost coefficient	Price of the feature	Price currently used	Discount %	Labor (€)
Matriisi Model	ROTATOR XXX	99.04	99.04	99.04	1	99.04	99.04	0	0
Matriisi Model	ROTATOR XXY	131.58	131.58	131.58	1	131.58	131.58	0	0
Matriisi Model	CRANE XXX	8968.21	5352.24	1736.28	1	5352.24	5352.24	0	0
Matriisi Model	CRANE XXY	5043.17	4874.86	4706.54	1	4874.86	4874.86	0	0
Matriisi Model	NO LIGHTS	0	0	0	1	0	0	0	0
Matriisi Model	LIGHTS	6080.1	3817.19	3062.89	1	3817.19	3817.19	0	0
Matriisi Model	**PLATFORM**	N/A	N/A	3230.43	1	0	0	0	

Figure 33. Defining prices for options

As shown in Figure 33, the average, minimum and maximum values for costs are presented while the user can provide discount percent, work, cost coefficient and finally the price. The price is the figure that will remain even if the costs of features alter. This way changes take place in average costs and in minimum and maximum values whenever the configuration knowledge is redefined while the price data remains the same. Thus, the price offered to the customer remains the same and simultaneously monitoring the costs changes related to price can be made possible. The price data entered by the user to the interface presented in Figure 33 is the data used also by the marketing configurator.

3.8.3. Revisioning configuration knowledge

For lifecycle purposes the task of revisioning the configuration knowledge is the main principle. The process for revisioning the configuration knowledge differs from the process of versioning and revisioning parts, components and modules presented in section 3.7. For configuration knowledge the concept of revision is only used. This is because the configurator only uses the revision number to determine the valid configuration knowledge. Even if there are different versions of configuration models the configuration rules are only revisioned in the databases. The different versions of configuration rules are established by the different intervals provided by the revisions established during the lifecycle of the product family. Thus, for configuration knowledge the versions of different configuration models can be determined through revision intervals of the configuration rules.

As mentioned many times, the configuration knowledge changes over time and when changes take place the configuration models become invalid, i.e. the configurations will not be concrete, valid and complete. To reach the state of valid configurations there must be carefully planned processes to update the configuration knowledge. Configuration matrices and the databases also have a role in this process.

Figure 29 provides insights into the changing environment of configuration knowledge. The changes from product design will affect the configuration models whenever the module interfaces change or more options will be introduced into the models. Also, when new revisions are established while keeping the existing module number the same, the problem will be the task of updating the configured products in the ERP system. This is considered later and the changes from product design are concentrated more closely here. Whenever there is a change related to the option structure or to module interfaces, the configuration models will change. This is actually one of the most critical parts that decide the flexibility of a configurator. If the configurator is only capable of presenting the current state of the configuration model, the problem is that going back to reconfigure with old configuration models can become fairly complex, i.e. configuration rules and knowledge become invalid over time. This is again a matter of the nature of configuration matrices that provide enough flexibility and means to tackle this problem.

Revisioning the configuration knowledge takes place every time the configuration knowledge generator is used. When the configuration models are changed and the application used, the revision will change every time. The latest revision of the configuration models defines the rules that are active when executing the configuration task. The start revision is declared to provide the information that is needed when reconfiguration is considered. If there is a need to configure a product with an old revision, the configuration rules active are the rules that have the revision number between the start and the revision fields. For example, if the active revision is 5 and we need to configure a product for after sales with revision 3, the configuration rules used will be the rules that include revision between the start revision value and the current revision value.

In order to determine the right revision interval, the configuration knowledge generator considers revisioning for all the tables individually. This is because the decision related to the revision interval will differ from table to table. For example, a new record for the production configuration rule is established when the code cannot find a specific rule representing the same machine, option, module, rule and revision minus one. If this can be found, the database is updated by updating the revision number, if not, the code will insert a new record to the database changing the configuration model for the configurator. Then again when the code is determining the revision for the pricing database, the code determines the need for updating by trying to find the same costs, machine, option and revision minus one. If this is not found, the code will insert a row with all the necessary information and declare the start revision to equal revision. This will start the lifecycle for this option and every time the cost for this option is altered, the code will update the cost and the revision number leaving the start revision unchanged.

The configuration matrices in Excel are the provider for all the configuration knowledge, but they are not necessarily needed to be revisioned. This is because when changes take place they are established in the Excel environment into the configuration matrices. When updating the databases with the configuration rules generator, the configuration knowledge will be revisioned and the previous model saved in the revision intervals. If the configuration model needs to be considered manually a reverse application can be used to provide the Excel presentation, i.e. not all the configuration matrices need to be available in Excel format.

When the configuration rules are revisioned there is also a possibility to analyze the cumulated data. When products are configured into the databases and configuration knowledge changes over time, the following analyses can be built:

- Cost development of options
- Cost development of sold machines
- Options available at different times and different configuration models
- Frequency of option sold
- Frequency of product design changes
    - options
    - modules
- Frequency of customer specification changes
- The number of options available in different versions
- The evolution of platforms during the life cycle of a machine

Next to the above mentioned analysis, the revisioning of configuration knowledge provides means to handle situations where configuration knowledge has been changed and there are needs to reconfigure an existing product that has not been manufactured yet. The following section will consider this situation more closely.

3.8.4. Algorithms to decide configuration knowledge validity for change situations

Configurable products experience different changes during their life cycle. While considering configuration knowledge, the changes that are critical are related to options and their dependencies between modules and themselves, module interfaces and their relationships. Thus, every time the generic product structure changes, the configurator databases need to be updated. The first problem here is the fact that there are many already configured products in the order books and when the customer requires a change to these configured products, the updated configuration knowledge could be invalid for the change process. It is possible to reconfigure with the revision of configuration knowledge that was valid when the product was configured in the first place. The problem here is that when module interfaces change during this time, some modules will be invalid and still included into the product structure because all the changes are omitted. The second problem is related to changes from product design when considering the order book configurations. When products are configured well in advance, there might be old revisions of modules as well as wrong modules in the product structure when production starts if the order book product configurations are not updated. To clarify, there are two main situations where the changes in configuration knowledge should be evaluated and they are defined as follows:

- When changing the customer specifications with the marketing configurator

- When changes take place in configuration knowledge that affect the configured product structures in order books

The latter situation suggests that whenever changes take place in the configuration knowledge, the configured products in order books should be updated, i.e. they should be reconfigured according to the valid configuration knowledge. The natural way of eliminating this factor is to design the products producible and robust enough so that these types of changes would not take place after the production ramp-up. The problem is that this is not the case for many reasons, and applicable processes need to be in place to be able to answer the changes that can otherwise produce harmful variation to the entire system. The prior situation for marketing can also be criticized when considering the latter situation. If the configured products in order books are updated every time the configuration knowledge changes, all the configured products would always be configured with the latest revision of configuration knowledge. This is important since when configurable products are designed carelessly the amount of reconfiguration can be quite high (order books of 100 machines for example) when considering frequent changes. This can be considered waste.

Either way, changes take place during the life cycle of the product and when the organization is just starting to use configurable products and a configurator, evaluation of configuration knowledge is essential when changes are established for both cases. The organization will use the same procedures for the above mentioned situations to define the validity of the configuration knowledge. When the processes work properly, the marketing side should always be able to change an order without a full reconfiguration, i.e. the change process takes care of the update task of configurations in order book. To check the validity of configuration knowledge, three algorithms are provided to establish grounds for systematic management of configured products in order books. The main duties of these algorithms are to:

- Decide if all the marketing options are present in the latest revision (see Appendix 2 for pseudo code)
- Decide if new options have been established and if so, consider their dependencies
- Decide if marketing configuration rules have been changed

The algorithms are in chronological order of execution. The first application checks that all the options are present in the latest revision of the configuration knowledge related to the customer specification. The problem here is that when a product has been configured and the customer has selected the options, these options have to be delivered. For many times options will be added to the configuration models and the problem of removed options is very unlikely to happen. There is still a possibility of this happening, i.e. it is included into the analysis.

The second application takes care of the new options that have been established. If these options have no relationships with other options, this stage will be passed. It is necessary to understand that a new option can have dependencies with other options while it can also affect the module selection of other options, i.e. the option combination of module selection is established. Next to this there is a possibility that other options affect the module selection of the new module provided by the new option. To clarify, this condition is checked by using the configuration rules for production. If the new option will not be found from any configuration rule than its own, it will have no dependencies. As for the other dependency of the other options affecting the selection of the new option's module, it is not necessary to check as long as all the options are present. This has been checked in the first phase. If all the options are present (phase one) and there are no dependencies between the new option and other module selection (second phase), the second phase is cleared.

The third and final application checks the marketing configuration rules. It is important to check the validity of marketing configuration rules because if changes have occurred in these rules, there are no ways of determining the new situation. Thus, there is a possibility that wrong options can be selected into a customer specification, i.e. the requirement of valid configurations will not be reached. This phase will only be passed if the marketing rules are exactly the same for the different revisions of configuration knowledge. The configuration rules generator will only insert the options with marketing rules into the database, i.e. if the option has no dependencies with other options, it will not be included into the database. This is a different situation from phase two since it concentrates on production configuration rules whereas phase three considers the marketing configuration rules. The entire procedure to check configuration knowledge validity is presented in Figure 34.

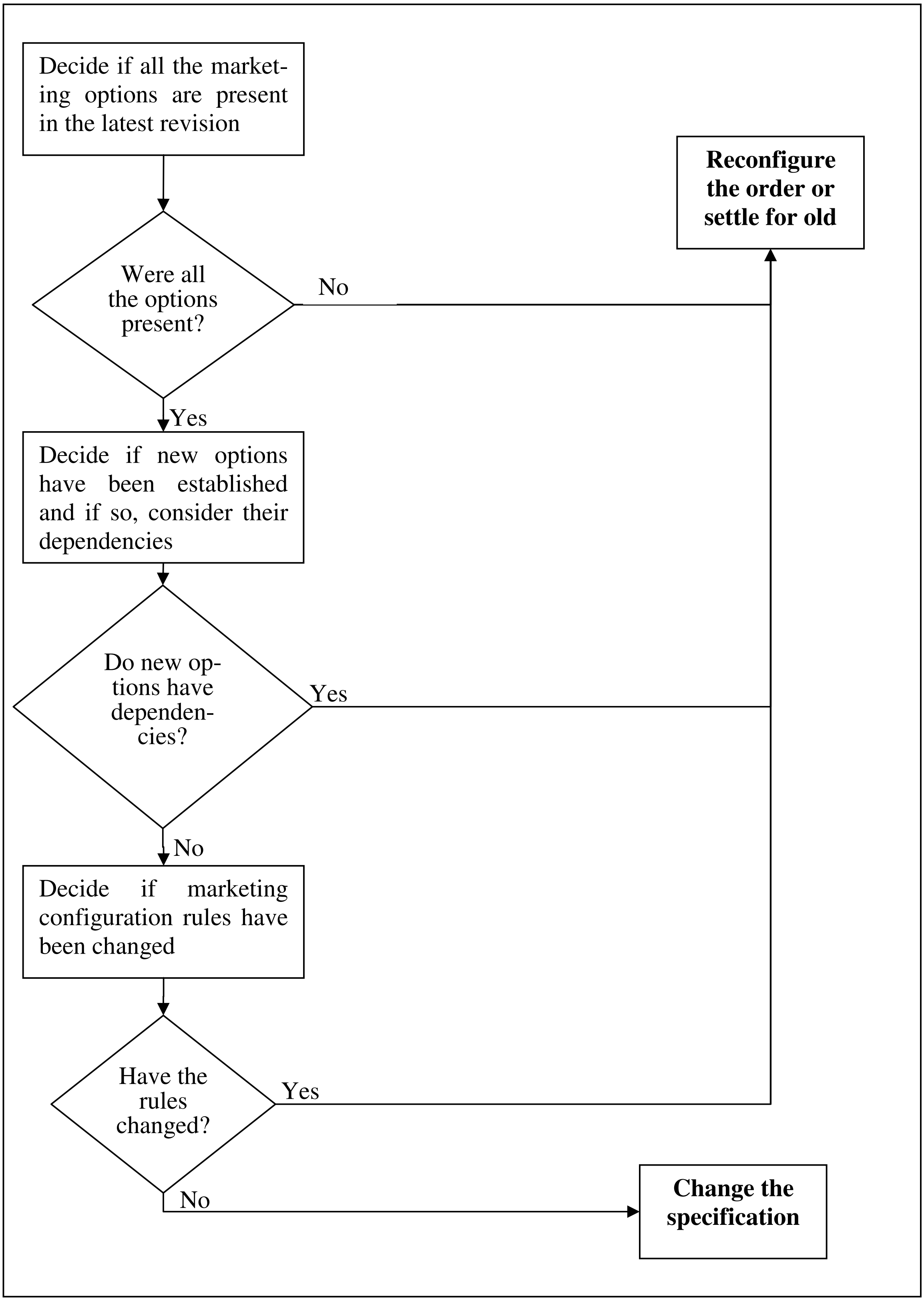


Figure 34. The flowchart for establishing the validity of configuration knowledge

All the phases have been presented in Figure 34 to make decisions about the validity of configuration knowledge. The outcome of the process is either to return the old configured customer specification and make the changes to the configuration with a new revision or to reconfigure the entire order if unacceptable changes are found. Using the three steps presented in Figure 34, there is the possibility to make the decision to retrieve the old configuration and to make the changes even if the configuration knowledge has been changed. The idea is that the requirements for this decision have been carefully considered and if one of the phases in Figure 34 fails, it will automatically require reconfiguration of the entire order. Another possibility is to settle for the old revision and make changes with old configuration knowledge. This will result in problems considering the validity of the configuration in the changed situation. It needs to be noted that the phases in Figure 34 can be used both for customer specific changes in the order-delivery process as well for product changes from product design. For customer specific changes passing the tests will enable partial reconfiguration whereas passing all the tests for product changes will enable automatic reconfiguration for the order book configurations.

As mentioned before, when the processes work as planned and the change process takes care of the updating task of order book product configurations, there is no possible way that customer related changes will have a problem when updating the order. Thus, all the orders in the order books will be configured with the latest revision of configuration knowledge. When the change process works, the process in Figure 34 will be the responsibility of the configuration knowledge maintenance process after changes in configuration knowledge have been established. The main benefit for the use of these algorithms is the fact that if the products in order books pass all the phases, they can be automatically reconfigured to reach an updated state while those that fail, will require reconfiguration of the entire order. When done properly, all the configured products will be configured with the latest revision of configuration knowledge and all the product structures in the system will have the latest module revisions.

3.8.5. Marketing configurator

The first part of the configurator is the marketing configurator that uses marketing configuration rules established by the configuration knowledge generator. The outcome of the marketing configurator is a customer specific option list that will be used later to configure the customer specific product structure with the production configurator presented in section 3.8.6. The emphasis in this study is not to concentrate on programming techniques, but to give an idea how the configurator works based on configuration matrices. The marketing configurator that has been established provides the following properties:

- Sequential configuration tasks according to the configuration matrices
- Pricing development as configuration task progresses
- Shows visually
    - selections that have been made
    - selections that have been omitted
    - selections that have been made unavailable
- Permits the user to go back by selecting the option that needs to be reselected

- Sends e-mail of the specification automatically to predefined stakeholders
- Provides order confirmation outputs
- Writes the valid specification into databases

The marketing configurator is based on the configuration matrices presented in this thesis. Configuration matrices have a predefined order of configuration task. This is the reason why sequential configuration is necessary. The marketing configurator uses the pricing data of options to provide the cumulative price of a configuration. The main idea is that while the costs of options are automatically defined, the options of the pricing software system presented in Figure 33 equal the options selected during the configuration task. The interface of the marketing configurator is presented in Figure 35.

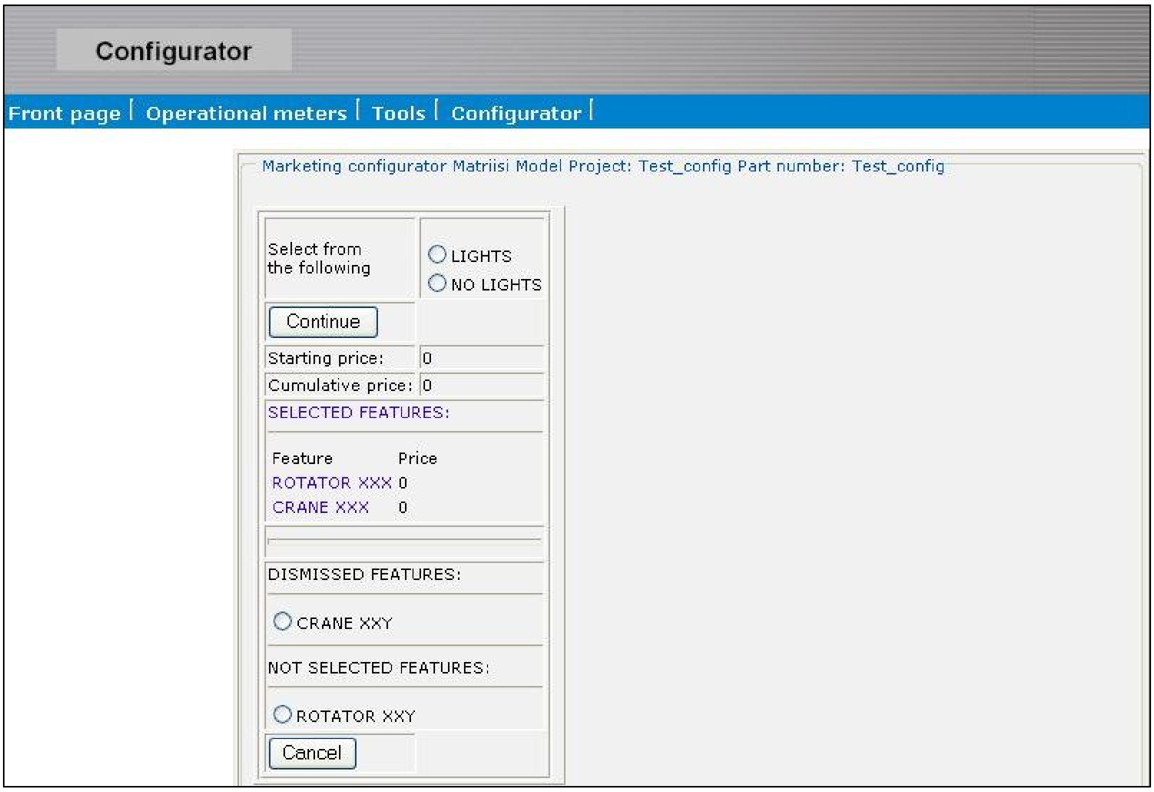


Figure 35. The interface for marketing configurator

As shown in Figure 35, radio buttons are used to select an option. When an option is selected, the “Continue” button is pressed and the configurator will provide the next feature and its options to be selected. Below the “Continue” button are the platform price and the cumulative price for the configuration. Below these features there are the selected options, waived options because of the marketing rules, and finally the options that were not selected. In the last two sections of the interface there are also radio buttons next to the waived and not selected options. The reason for these buttons is to provide means to go back and change a selection of a feature. By selecting an option from either of the two sections and pressing the “Cancel” button will take the configurator back to that feature and show its options in the right hand corner. After this the configurator moves sequentially and the subsequent options must be selected.

Finally when the configuration is ready, the configurator will provide the order confirmation and ask for the user to confirm the configuration. After this confirmation the e-mails are sent automatically to the stakeholders that have been defined by the user. This phase is shown in Figure 36.

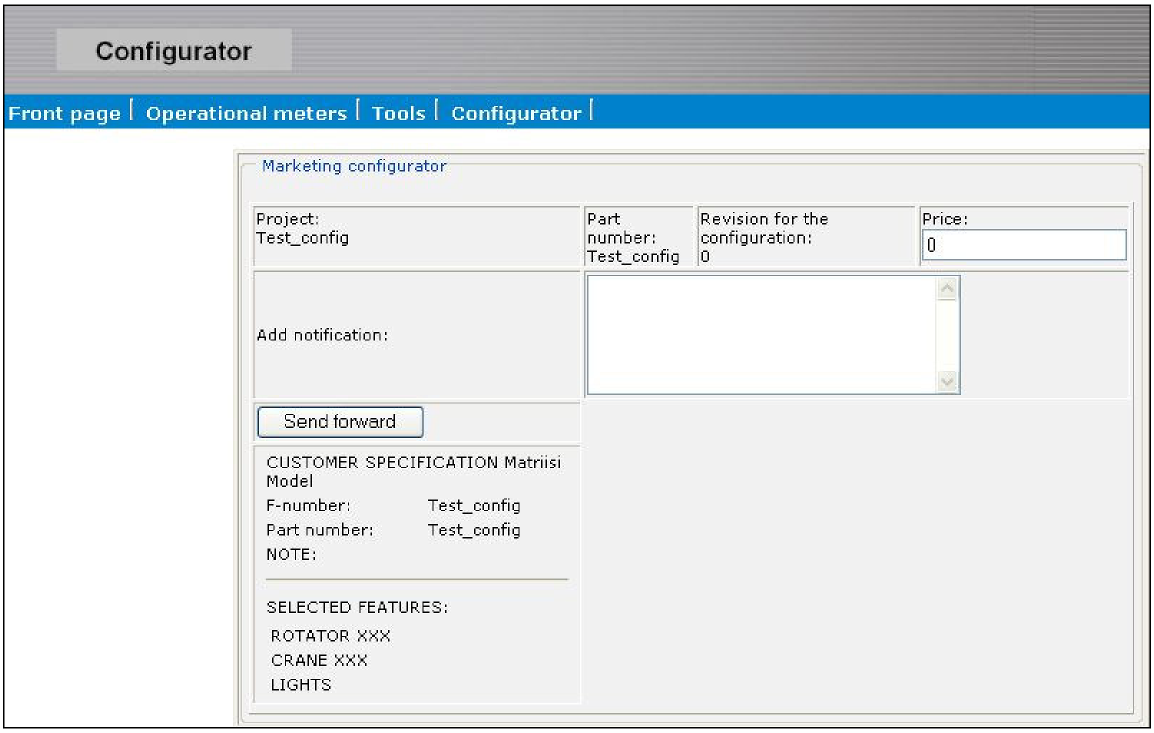


Figure 36. Configuration confirmation

There is also a field where the user can insert any abnormal wishes. This field is provided because if the customer wants something that is outside the configuration knowledge there should be a place to insert this data. This data will be automatically inserted into the e-mail that will be sent after pressing the “SEND FORWARD” button. The subsequent phases of the configuration process will take care of processing the extra data.

The main goal for the marketing configurator is to provide valid information to the subsequent phases. As configuration matrices provide all the required knowledge for the configurator, there is no information that can come aside from the matrices. If there are more options needed, they need to be designed by product design and then configuration matrices and the databases need to be updated. After these operations the configurator will be able to show new features and their options. As shown in this section, the marketing configurator is totally dependent on the quality of data in the configuration matrices. For this reason the processes to maintain the configuration knowledge as well as to design new features and their options need to be systematically handled and well understood.

3.8.6. Production configurator

Next to the marketing configurator that provides the customer specifications there has to be a way of converting this specification to the customer specific product structure. This will be done by the production configurator. The entire system is totally synchronized with the configuration matrices and if there needs to be changes, for example, to the description of options, it is done to the configuration matrices. After this the databases are updated and the configurator is ready to be used. Consider the implications though. If there needs to be changes to the configurator and only one description of an option is changed, the checking procedure presented in section 3.8.4 will not pass even the first phase. This means that automatic configuration of the customer specific product structures by the production configurator cannot be used and if the structure needs to be updated, the order has to be reconfigured. This is the situation for customer specific changes also. Note that whatever happens, the two configurators, marketing and production, need to use the same revision for both configurations. This implies that even if the configurator is flexible, the need for systematic processes is very apparent in order to eliminate the needless efforts related to the changes.

The way the production configurator works is that the specification of the configured order and product specific production configuration rules are retrieved from the databases. After this the configuration rules are compared with the specification. Notice that the entire system works with AND/OR operators only. The configurator engine can be very simple to perform this task. It only needs to split up the configuration rule considering the operators and then check if the appropriate options are selected into the specification provided by the marketing configurator. For example, if we have a module XX with a production configuration rule such as (Option_1 OR Option_2) AND (Option_3 OR Option_4) AND (Option_5) the configuration engine will split the rule as follows (Table 6):

Table 6. Splitting up the configuration rule

Feature_1	Feature_2	Feature_3
Option_1	Option_3	Option_5
Option_2	Option_4	

As shown in Table 6, the configuration rule is decomposed into features. If the configuration matrices are considered, the OR operator will be selected between options for the same feature, i.e. the gray color in the configuration matrices does not change between these options. Then again, AND operator needs the color change, i.e. the feature is changed between the options. Using this idea the production configurator can come up with the following table that will be used to determine if module XX will be included into the customer specific product structure (Table 7).

Table 7. Table for interpreting the module inclusion

Option	Found
Option_1	True
Option_2	False
Option_3	False
Option_4	True
Option_5	True

According to Table 7, module XX will be included since all the required options were found from the customer specification. Options 2 and 3 were not found, but because they both have an alternative option 1 and 4 respectively selected, all the requirements for module XX have been satisfied. The example presented above will iterate through the entire database of the production configuration rules and by doing this the customer specific product structure is defined for the configured specification. This product structure is stored into the databases of the configurator and is now ready to be moved automatically to the ERP or PDM system if integration between the systems exists.

The production configurator is separated here from the marketing one, but in reality there is no reason to keep these two configurators separate. This is because nothing special happens after the customer specification has been defined. Considering the configuration matrices and their nature, the two configurators should be integrated and the customer specific product structure would be ready right after the marketing configuration has been confirmed. This is a matter of processes and also a matter of changes that take place during the order delivery process. If the processes and the products are very robust in nature, the configurators can be integrated whereas when changes occur frequently and processes need human intervention, the two should be separate.

3.8.7. Making changes to the customer specification

An additional tool has been provided to supplement the marketing configurator in order to make changes to the marketing configurations. As changes for customer specifications are inevitable, there needs to be a way to handle this situation. First of all this phase needs to communicate with the ERP system because not all the configured products can be changed. This means that from the ERP system only the product structures or configurations that are in the order book or released to the production can be changed. It would be preferable to include only the products in the order book since changing the specification of a released order is questionable. This is again a matter of company specific processes concerning this situation. The features of this application are as follows:

- Provides the products that are in the order book or released
- Checks the validity of configuration knowledge
- Returns the entire configuration with marketing configurator
- Provides the old specification to aid reconfiguration
- E-mails sent include only the change
- Production configurator provides the changed situation for the module level

The product that needs to be changed can be selected as shown in Figure 37.

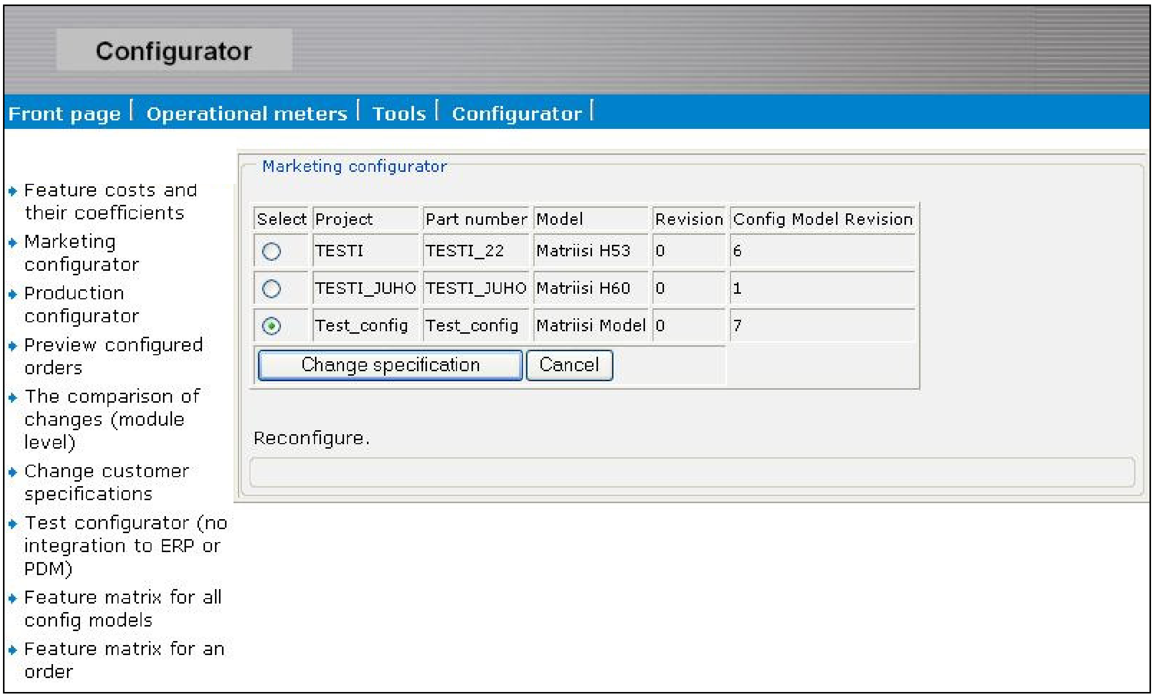


Figure 37. Selecting the product that needs to be changed

When the product is selected and the process is moved forward, the application will check the validity of the configuration knowledge, i.e. the configuration knowledge used at the time of configuration is compared with the latest available configuration knowledge. The three phases described in Figure 34 are considered. As mentioned before, the configurations in Figure 37 should always be done with the latest revision of configuration knowledge, i.e. the checks should be passed every time. If the checks are passed, the latest configuration knowledge is used for the product and the final state of the marketing configuration is returned as shown if Figure 38.

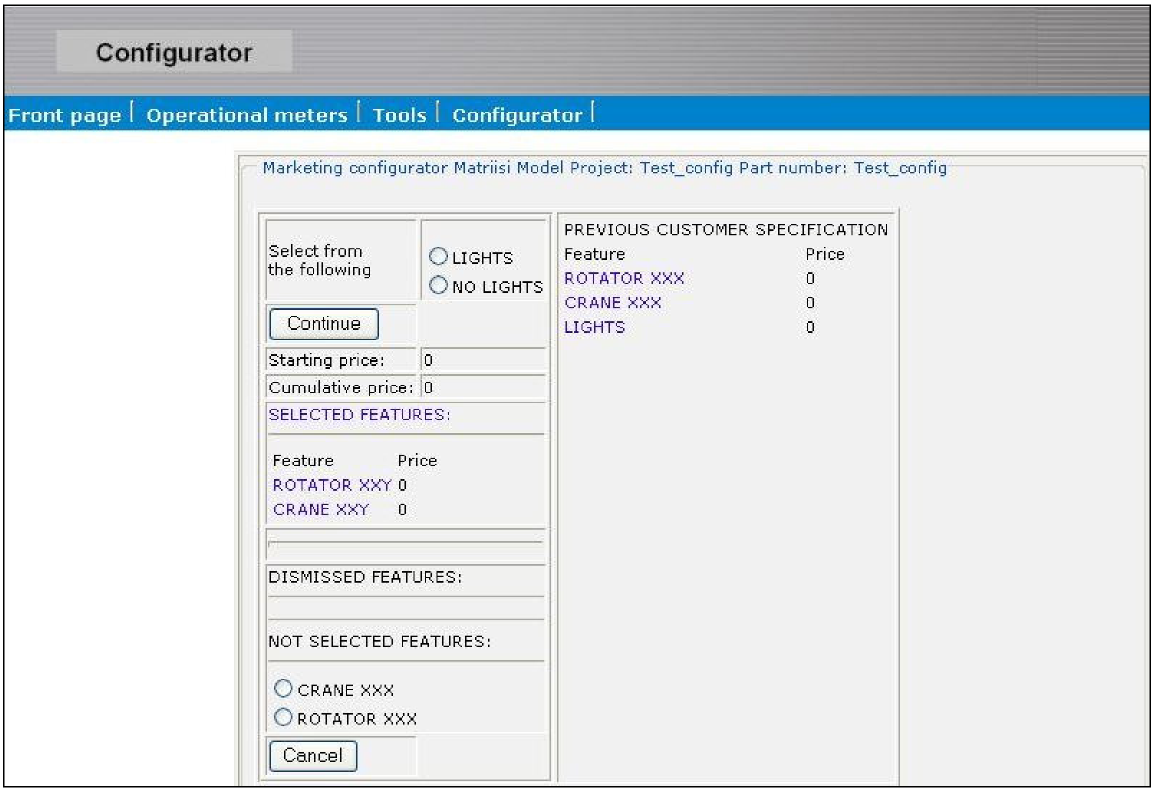


Figure 38. The returned state of a marketing configuration

As shown in Figure 38, the currently valid specification is displayed ("PREVIOUS CUSTOMER SPECIFICATION"). It needs to be remembered that we are considering a sequential configuration model. This means that when we select one of the options from either of the two lowest sections and select "Cancel", the configurator will take the user back to the selected option and the configuration needs to be completed from there by selecting the wanted options that follow. The worst case is that the option needed to be changed is the first option. This will require selecting all the options again to complete the reconfiguration. Note that the specification in Figure 38 is of great help when reconfiguring the product.

When the new configuration is confirmed, the e-mails will be sent again to the predefined stakeholders. These e-mails include only the information about the changed options. Figure 39 provides an idea of the type of e-mail that will be delivered.

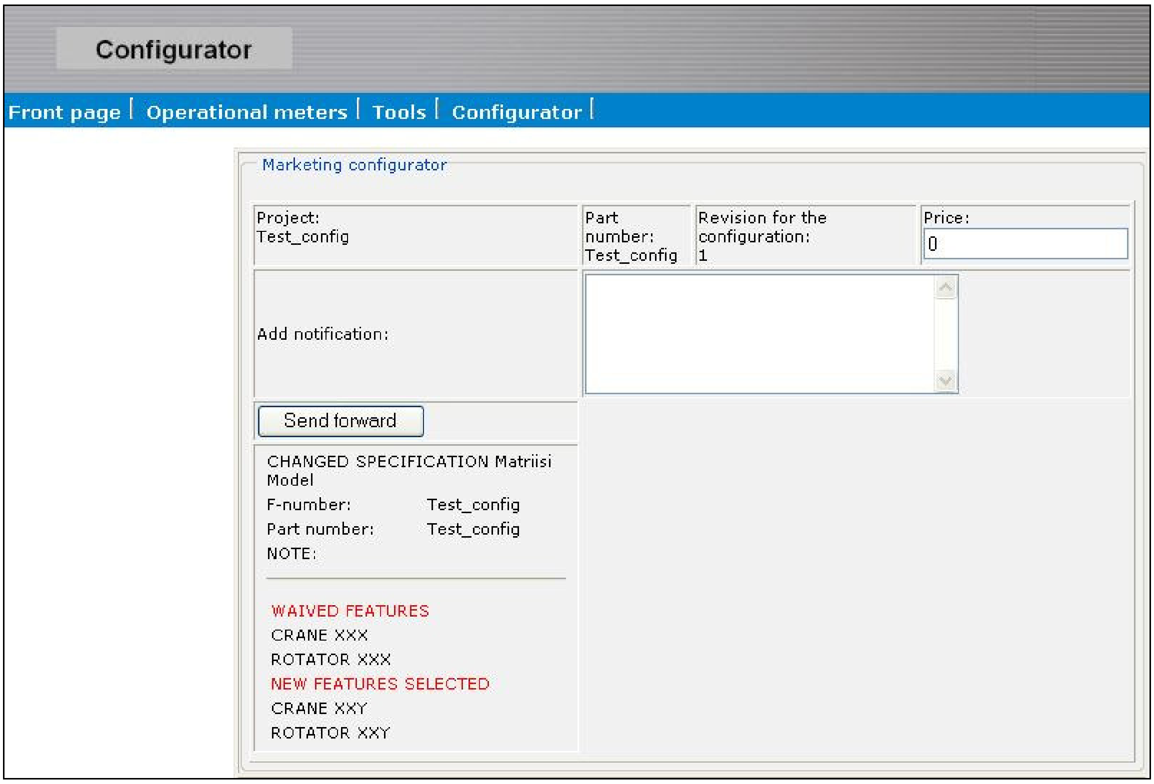


Figure 39. Confirmation of the change

As shown in Figure 39, only the changed options will be displayed. For this example there were two features changed and there will be two options displayed in both section, waived options and added options. This is the information that the organization needs when customer changes in the order-delivery process take place.

Considering the process further, the production configuration needs to be accomplished for the new specification. If there is a good integration between the configurator and the system where the customer specific product structures are stored, the only thing to do is to replace the structure with a new one. With the integration of these two systems, this can be done automatically. On the other hand, if no integration exists, there needs to be a way to present the changes in the module level. Figure 40 shows the result due to the change done in Figure 39.

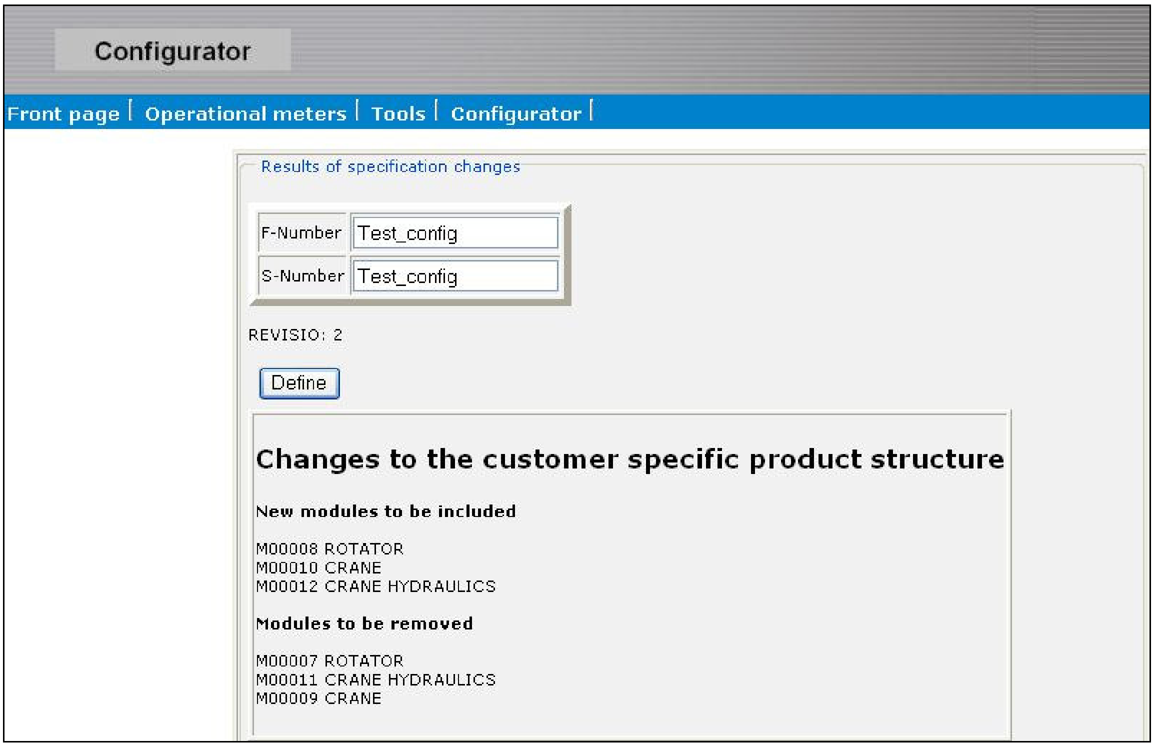


Figure 40. The change to the module level

As shown in Figure 40, there will be three modules added and three modules withdrawn from the product structure. This is now more easily done to the system (ERP/PDM) because the entire product structure needs not to be considered. Naturally the complete customer specific module structure is stored into the databases of the configurator.

3.8.8. The processes of making changes to configuration knowledge and to configured products existing in order book

This section will conclude the ideas presented in the above discussion about the changes in the configuration knowledge and customer specifications. As mentioned before, changes that the configuration knowledge experience are related to changes from product design and the changes that the configured product structures experience are related to customer specific changes during the order-delivery process. Both of these situations are critical to the concept of the configurator. Figure 41 presents the process for changing the configuration knowledge when product design causes variation.

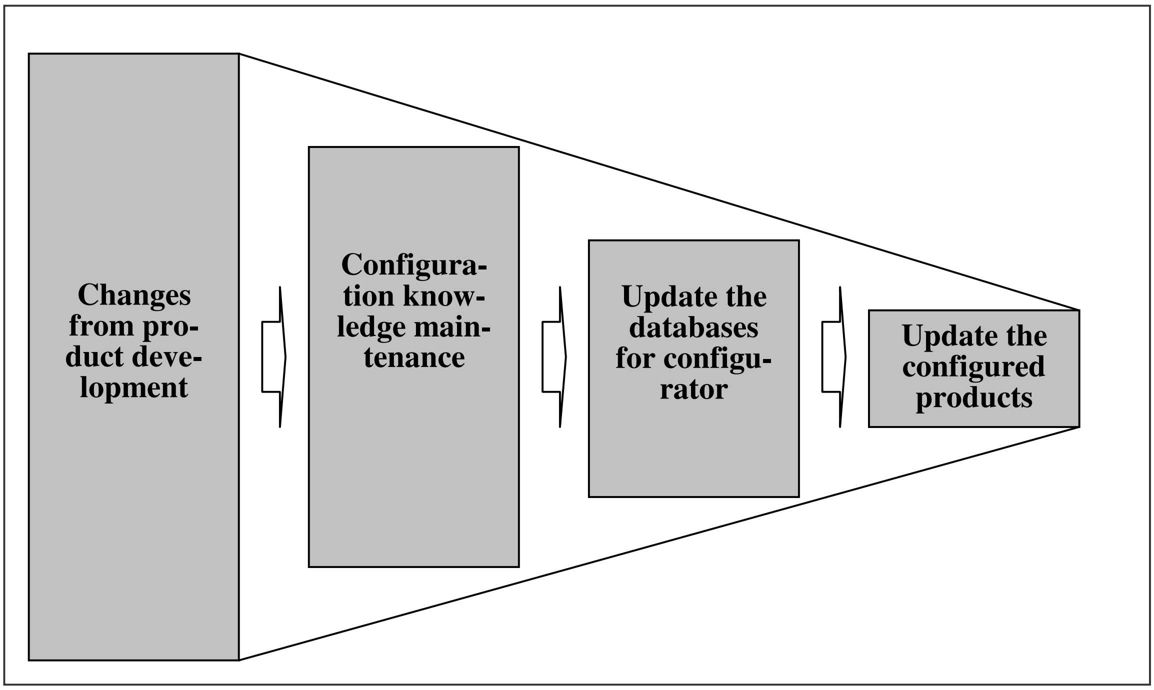


Figure 41. Updating the configuration model when product development changes the product

The way the process in Figure 41 works, is that whenever product design produces changes to existing products that change the configuration knowledge, the configurator databases must be updated. When the change requisition arrives in configuration knowledge maintenance, the module interfaces are checked and the configuration matrices are updated accordingly. This can be very tedious and time consuming if modularity for some reason is not in adequate levels. For example, when the configuration models are maintained somewhere else than in product design, lots of iteration concerning the module interfaces can occur. This is again a lack of process capability, and in that sense very relevant. When the configuration matrices have been updated the configuration generator is used to update and revision the databases. When the databases are updated, the configurator uses the latest revision of the configuration knowledge automatically. The only thing that needs to be checked is the pricing since when new options are introduced, the prices related to the new options are manually inserted as described in Figure 33. When this is done, the configured product structures from the order book need to be updated. For this, the algorithms provided in Figure 34 will be used to determine the list of products that have to be manually reconfigured. Others that pass the gates in Figure 34 can be automatically reconfigured with the production configurator. Furthermore, when adequate integration with a system that stores the customer specific product structure is available, new module structures can be automatically uploaded to the system. If there is no integration, the tool shown in Figure 34 must be used to manually update the module structures. When all of this is accomplished the following state has been reached:

- Updated configuration models
- Updated configurator databases

- Valid configurator
- Valid product configurations in order book
- Updated product structures and module revisions (if integration between the configurator and ERP or PDM has been established)

The last issue is important when scheduling module revisions. When modules are in the ERP system, updating the product structures is critical when modules are revisioned and scheduled. When the product structure is inserted into the system, the module will use the latest revision available according to the scheduling. When the product is configured well in advance module revisions can change making the module structure invalid. The point is that when revisioning modules for future use, the configuration models will not change. The main thing is that the product structures in the systems should be updated. This is possible when integration between the configurator and the system exists. The entire order book can be reconfigured automatically by the production configurator and new product structures can be inserted into the system, i.e. all the manual work is eliminated while making changes possible. Note that in this case all the checks of configuration knowledge validity will be passed.

The next possibility for a change situation is the customer specific change during the order-delivery process. This change process is illustrated in Figure 42.

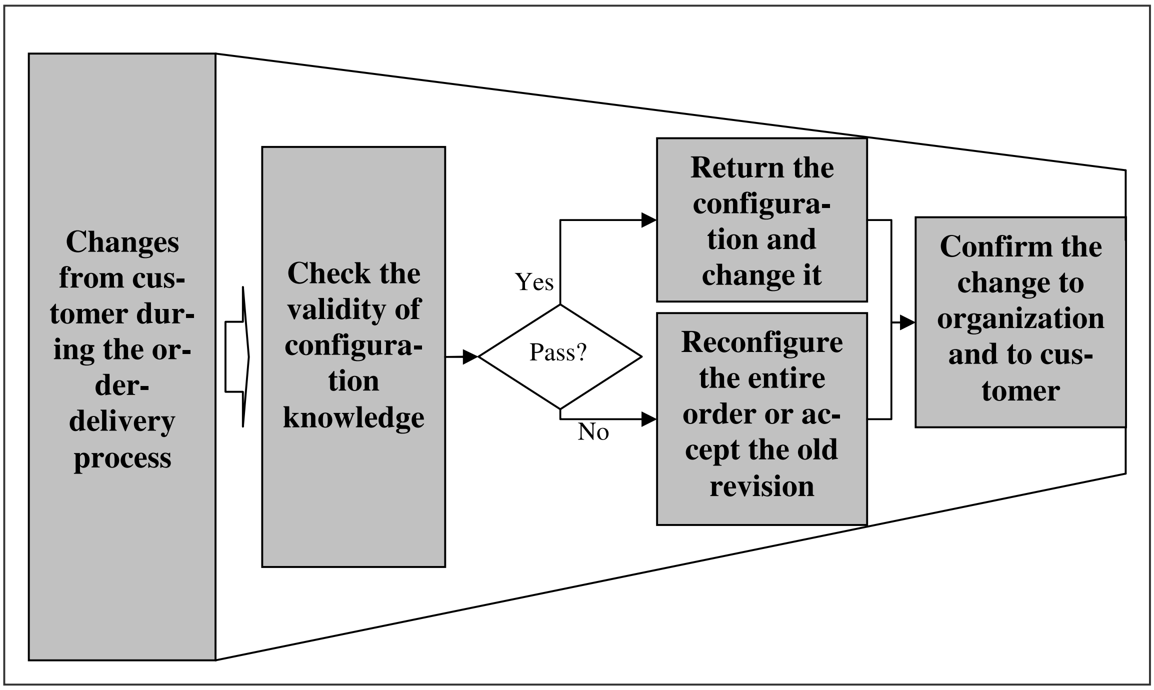


Figure 42. Customer change during the order-delivery process

The core for the process presented in Figure 42 is the validity check for configuration knowledge. As mentioned before, the validity check should always be passed when the process in Figure 41 works as planned, i.e. all the products in the order book should be configured with the latest revision. When processes are not working properly, lots of options will be introduced and many module interfaces will be changed resulting in a situation where product offered to the customer at the point of sale can be different from the final product produced. This is the reason why for a configurable product the change process next to the product design process should be very systematically handled. This is the also the reason why configuration matrices are tightly linked to the process of MBI presented by Aarnio (2003). Following the process presented in Figure 42, after the confirmation of change has been established, the production configurator will take care of the product structure update. Again, if there is adequate integration, this update is automatic and if not, the interface shown in Figure 34 will be used to determine the new situation. The following state is reached:

- Customer satisfaction secured
- Changed order
- Information automatically to right stakeholders in the organization
- Updated product structure
- Latest revisions of modules (if integration exists)

For this process it is needed to mention that only one order is handled and it is much simpler than the process of making changes to the configuration knowledge. What is important though is the fact that when using the process presented in Figure 42, there is a flexible and systematic way of making changes to existing orders. This is mainly possible because the configuration databases are revisioned. This type of approach makes a configurator possible even in an environment where changes are frequent both for configuration knowledge and customer specifications.

3.9. Closing configuration matrices and the implications to organization and its processes

The idea of configuration matrices is to visually and systematically present the configuration knowledge related to different parts of the organization. As noted, the main idea is to model, document and present the knowledge related to configurable products. Surprisingly, the information related to configurable products can easily be analyzed and used in various locations in the organization to enable effective means to make processes of many kinds more effective, quicker and disturbance free by offering various kinds of software tools. Also the modularity needs to be seen broadly to satisfy the needs of many stakeholders inside and outside the company. This requires broad understanding of the current situation of modularity as well as a vision of how modularity should be developed, i.e. it is too easy to solve the problem of modularity by considering one function of a company and the problems occur when modularity is considered broadly. Configuration matrices together with the evolution model of modularity (Lehtonen et al. 2003) and real life offer means to systematically develop product modularity into a wanted and reasoned direction considering the entire aspect of producibility.

The power of well-defined configuration knowledge does not affect the configuration process only. The effects can be seen as steps are taken to handle modularity holistically considering the entire organization as configurable products make customization of products possible. Even if modularity does not solve all the possible problems of the company, it still needs to be considered very carefully. While considering configurability and the related knowledge, the systematization of this type of knowledge is imperative. Configuration matrices and tools related can offer many opportunities for companies to understand and also to analyze their current level of modularity next to the operative analyses helping daily life. While the configuration matrices are maintained and product structures are designed considering the wanted level of modularity, the results of the analysis produced by the tools generate accurate data useful for many stakeholders in and outside the company.

Finally, the change processes for configuration knowledge and configured products were introduced. The configurator established considering these processes and the features of configuration matrices provide effective means to handle the configuration process in a changing environment. The main issue is that no manual labor is needed to update the configurator databases; the configuration knowledge generator makes this automatically while simultaneously considering the revision of the configuration knowledge. The configurator also provides pricing for marketing. The difference lies in the fact that the cost can be automatically defined by an autonomous configurator integrated into the pricing software system.

4. CASE STUDY

Ponsse Oyj was the case company for this thesis. For Ponsse there were needs to further develop modularity using an outside resource and the task was to reveal the problems of the current situation. While getting an idea of the current state by using the DSM approach trying simply to figure out the dependencies within a product family, the idea of configuration matrices surfaced. As shown in the following sections, the use of these types of configuration matrices has helped Ponsse to understand its situation related to product structures and also to carry out extensive projects for many parts of the organization.

As the research in Ponsse was carried out, the literature review revealed many aiding thoughts and the most important for Ponsse was the evolution model presented by Lehtonen et al. (2003) next to the ideas of MBI and configurability presented by Aarnio (2003). All of the elements, excluding Dymo, were noticeable in the case company and for the same reasons as presented by Lehtonen et al. (2003).

Using the evolution model, configuration matrices and reality there was an additional Masters thesis launched to decompose the existing product structures using the ideas presented in this thesis and also to re-engineer the product development process to meet the needs for the new situation. The results of this Masters thesis have been promising as the designers get a grip of modular engineering as a way of working. The main result from the Masters thesis was the disintegration of the cabin structures with the help of configuration matrices. The cycle for development of modularity for Ponsse has been as follows:

- The current state of modularity had shortcomings in several areas in the organization
    - research project launched concentrating on the aspect of modularity
- Revealing the product structure by the use of configuration matrices enabled Ponsse to understand the level of modularity company-wide
- Studying the configuration matrices and their effects to the organization made the usability of matrices clear for many processes
- While understanding the importance of product structure and its modularity the effort has been targeted to product development to produce the wanted modularity

As noted above, the main issue to tackle during this research was to further develop modularity. It turned out that the concept of configuration knowledge provides a powerful framework for the company when configurable products are used. In the following text the results of the case study are reported. As shown, and as discussed previously, configuration knowledge does not only provide the means for the configuration process, but can be widely used around the organization. The case study has been divided into analyses of the current state (section 4.1), configuration matrices and the base machine estimations for the case company (section 4.2) and results experienced by the case company (section 4.3).

4.1. Analyses of the current state

Current state analyses include the presentation of the case company Ponsse Oyj and its products in sub-sections 4.1.1 and 4.1.2 respectively. Sub-section 4.2.3 presents modularity of the products and product development process while sub-section 4.2.4 considers the configuration process of the case company.

4.1.1. Ponsse Oyj

Ponsse manufactures and markets forest machines for the cut-to-length method of mechanized logging as well as wood harvesting-related information technology. Ponsse produced 399 machines in 2003 resulting in a turnover of 163.5 million euros. Currently Ponsse employs 553 people located in different countries. The Ponsse Group consists of the parent company Ponsse Oyj in Vieremä, Finland and the subsidiaries Ponsse AB in Sweden, Ponsse AS in Norway, Ponssè S.A. in France, Ponsse UK Ltd. in the UK and Ponsse USA Inc. in the USA.

4.1.2. Case products

Machines produced by Ponsse can be divided into harvesters, forwarders, harwarders, cranes and harvester heads. In addition to these products, information technology is heavily involved. Harvesters take care of the cut-to-length method, forwarders collect the harvested wood, and harwarders are combination machines able to perform both tasks. Main products are presented in Figure 43. Ponsse has two harvester machines Ergo and Beaver and four types of forwarders: Buffalo, Buffalo King, Wisent and Gazelle. For the product family of harwarders, Ponsse has Buffalo Dual and Wisent Dual.

Figure 43. Harvester, Forwarder and Harwarder (courtesy of Ponsse)

Next to the above machines, one of the most challenging product groups is harvester heads. Figure 44 presents the different harvester heads available. From the upper left corner the first product is the smallest harvester head H53. Then the following four pictures present the harvester head family 60 with models H60, H60E, H60BW and HW60. Finally, the last picture in the right hand corner presents the biggest harvester head H73.

Figure 44. The family of harvester heads (courtesy of Ponsse)

Considering the concept of configurable products and modular system presented by Stake (1999, see 2.2.3), all of the above presented products have their own configuration matrix to present the configuration knowledge related to the product families. Thus, all the modular systems have a configuration model of their own. The cranes are the easiest to configure whereas harvester heads are very complex due to the large amount of features having dependencies between each other needed to be selected during the configuration process.

4.1.3. Modularity and product development process

Products in the case company have been divided into harvesters, forwarders and harwarders as presented above. Harvesters and harwarders are decomposed into basic machine, crane and harvester head whereas forwarders are decomposed into basic machine and crane. All of the three entities for the harvester are configurable, i.e. they form their own modular systems with generic product and feature structures. All of the entities discussed have their own configuration models and they are configured separately and produced with their own shop orders. The product structure of all the configurable modules are decomposed into a functional structure as presented in Figure 45.

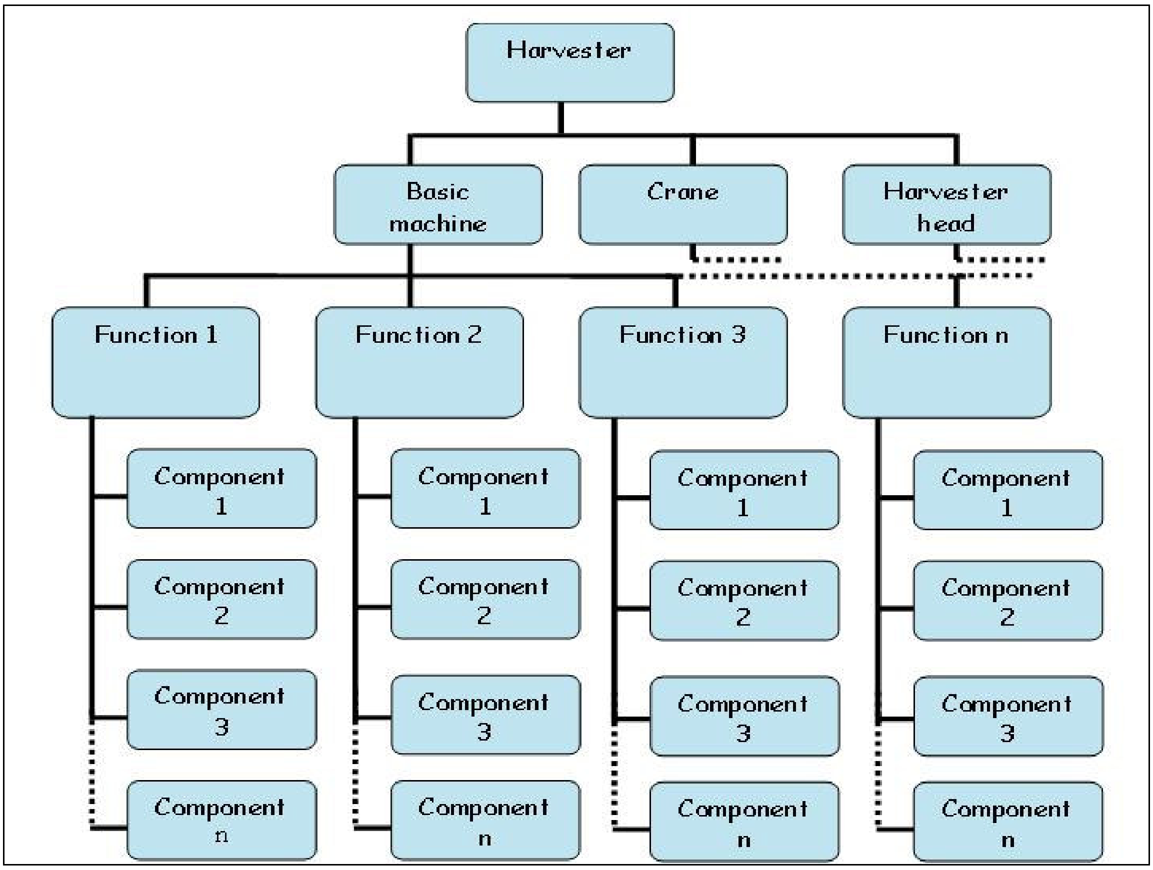


Figure 45. The current product structure for the case company products

The same idea of functional structure is used in every product. From now on we will concentrate on one harvester machine including one basic machine, one crane and one harvester head. As shown in Figure 45, the entire product structure is decomposed into functions which then are directly decomposed into parts and components. As mentioned many times, production cannot work with functional structures. This problem is serious when final assembly is considered, i.e. production cannot build functions and then assemble them.

If functional modularity is decided to be used, the similarity between production system and product structure needs to be secured. One solution is to use features offered by ERP systems, i.e. routing all the parts and components in the functions to be able to have meaningful entities to work with. As the customer structure has been established from customer specifications in the form of functional modules, the parts are routed into the production by using module and routing knowledge provided by the ERP system. As mentioned before, the problems of these types of product structures occur in production especially in final assembly. For Ponsse the structures below the functional level are standard and the drawings for machining and welding are naturally available. On the other hand, there is no documentation for assembly of any kind provided by the product design other than spare part documentation.

The configuration task takes place in the functional level of the product structure for every main entity. A modular system for every entity has once been designed considering mainly the order process (configurability), i.e. the emphasis has been to serve the customer as efficiently as possible. Production and after sales work with what they get since the principle for the case company has been to avoid multiple structures in the organization. For the case products the first step to modularity was the definition and creation of functional modularity which altered the way of working as well as the organization for product design. The problem is that if the organization and the processes are not considered deeply, sub-optimization can occur while developing new designs. The current product design process for the case company is presented in Figure 46. The concentration is on the product structure, i.e. when is the product structure ready for other parts of the organization.

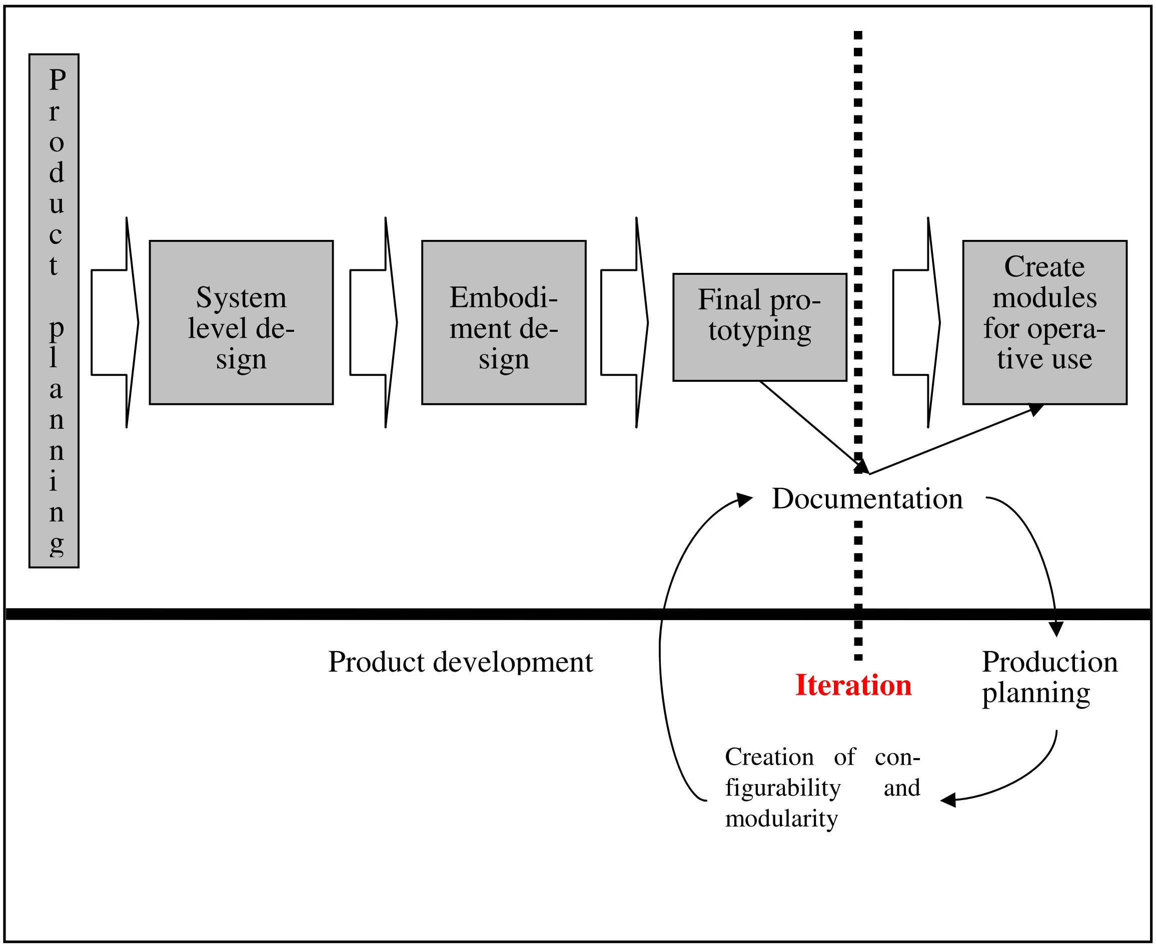


Figure 46. Product design process for Ponsse

From the beginning, the use of functional modularity was developed mainly by the production function since the need to easily configure customer specific product individuals was considered to be the most important issue. This actually enabled the case company to very effectively use modularity in order to satisfy customer needs. As seen in Figure 46, the product design process was left with no responsibility of modularity considering product structure. While the products are configurable, it is clear that product development designs the needed variants for product structures, but the decomposition of the entire product is mainly done in co-operation with the designer and the personnel from documentation. The output from the documentation department is spare part drawings. From these drawings production forms modules while the emphasis is to have the same interfaces with the spare part drawings, i.e. lots of iteration between production and documentation is usually needed if there are problems with the module interfaces. The conclusion is that even if the processes for Ponsse's product development are effective and flexible, the problems generating modular structures have effects that cause variation all over the organization. The problem is the late formation of the module structures to be used in the production and also the formation of the modular architectures in general.

4.1.4. The current state of configuration process

The current situation of good configurability of products has been the case for Ponsse for many years. The configuration task is executed manually after many phases including marketing, financing and sales during the order process for a sold machine. The nature for this type of business is the changing customer needs and expectations concerning the final product during the order process, i.e. the customer specific product structure alters and causes variation mainly to production if the changes take place very late considering the due date for the product. Basically the steps to fulfill an incoming order from sales representative considering important issues in the context customer specific product structure are as follows:

- Receive incoming order
- Check the validity of the order
- Setup a new sales order in ERP
- Setup a new PDM project for the order (use the sales order number from ERP)
- Copy a predefined preliminary product structure into the PDM project and generate new product id-numbers
- Tab option codes into the sales order in ERP
- Send e-mail to various parts of the organization to secure that all the important stakeholders are aware of the situation
- Generate the customer specific product structure

The above list of tasks during the configuration process includes many functions and people and iteration in changing environment is very likely to occur. Predefined preliminary product structures are for forecasting purposes and used for the first structure when setting up the PDM project for a new order. The forecast, i.e. a list of products to be sold, is updated with this preliminary product structure with a new product id-number by replacing the old product located in the wanted delivery date with the new product id-number. While the process advances, finally the production planning personnel uses the preliminary product structure and their experience to define customer-based product structures that should be the final structure for the production to use if no changes take place. This product structure is generated by taking modules from and putting new modules into the preliminary product structure according to the customer selected options. The configuration knowledge needed to carry out the configuration task was basically documented into the experts' heads and into a few sheets of Excel making these experts very important for the company. Manual configuration has been traditionally made in the PDM system and the transfer into the ERP system has been automated by integrating the two systems.

In order to get an idea of the process times the following chart presented in Figure 47 was defined by analyzing incoming orders.

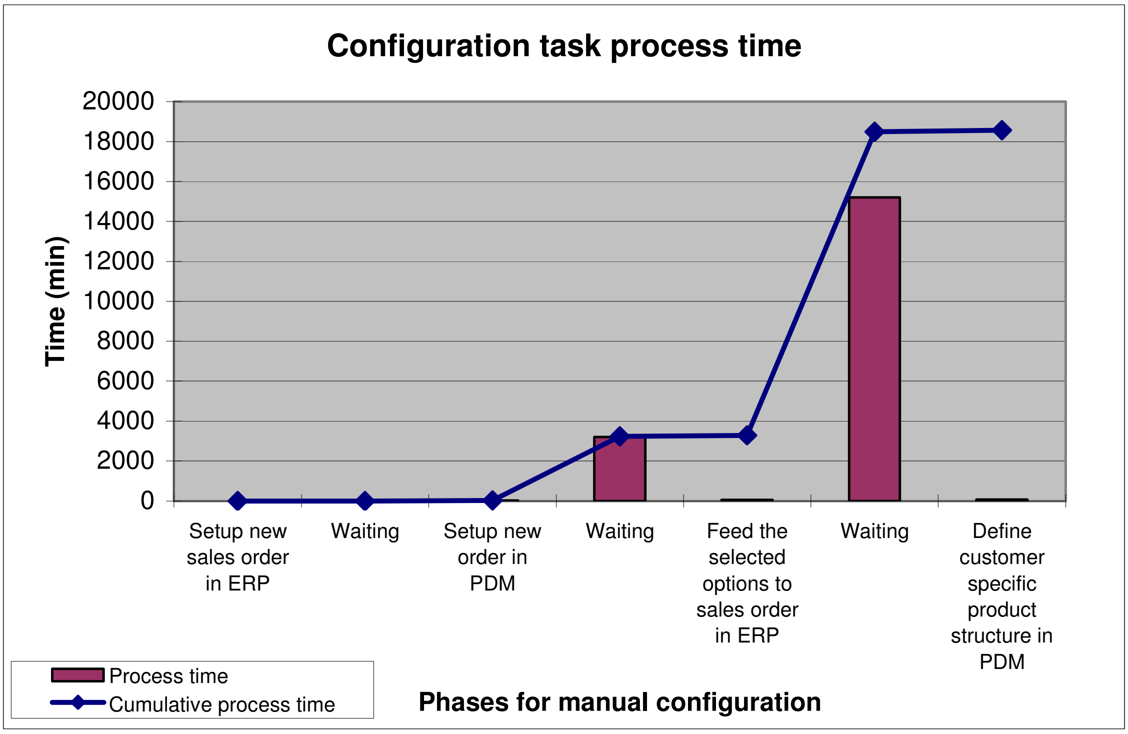


Figure 47. The process time versus waiting time in the current situation

As seen in Figure 47, the total process time for selected orders (N=39) is approximately 170 minutes while waiting time is over 18,400 minutes making the relation of process time per waiting time to equal 92.3%, i.e. from the total process time waiting takes over 90 percent of the time.

One of the problems mainly causing long waiting times is the fact that much of the tasks are executed in different locations of organization and communication between these different locations is not necessarily at its best. The problem is also that there are different pieces of processes formed along the way and information does not flow smoothly enough through the organization. The order handling process is very flexible and manually configuration task can be accomplished by this process virtually in any circumstances. The problem is the variation inserted into the system when accepting incomplete specifications from the sales department.

4.2. Configuration matrices and the base machine estimations

In this section some statistics are shows of the developed configuration matrices. Sub-section 4.2.1 reveals the developed configuration matrices and sub-section 4.2.2 considers the means how the platforms for the configuration matrices were confirmed through base machine estimations.

4.2.1. Developed configuration matrices

As presented above, one of the problems for the case company was the process of modularization as well as the level of documentation of the configuration knowledge. The concept of configurable products for Ponsse is a strategic approach to fulfill customer needs while simultaneously considering modularity in a broad sense to be able to develop effective processes company-wide (see section 4.3).

Configuration matrices triggered many clarifying aspects in the context of understanding the level of modularity of the products. As all the products for Ponsse are configurable, a total of 22 matrices were needed in order to cover the entire spectrum of products. These 22 configuration matrices were developed by using the approach presented in section 3.4.2 together with configuration experts of the company. For these configuration matrices there were total of 2,029 modules while 1,063 modules were different from each other. These 1,063 modules included a total of over 18,000 parts and components, i.e. there were over 18,000 parts and components in assembly integrated into the functional modules. From these 18,000 parts and components 4,186 were different from each other. These figures are in the context of configuration matrices, i.e. they consist of only final assembly parts and components. There were a total of 545 features in these 22 configuration matrices. As mentioned, the basic machine, the crane and the harvester head are configurable entities in a sold machine. Basic machines, cranes and harvester heads have an average of 51, 7 and 29 features to choose from respectively. There were a total of 1,585 options in these 545 features.

Considering the benefits that the case company experienced with the configuration matrices developed, are as follows:

- Systematically documented configuration knowledge
- Understanding of the current level of modularity
- Inputs for product design to decompose the platform differently
- Generic product structures
- Increased quality of configuration knowledge as matrices were further developed
- Connection between the modularity and production
    - late point variation
    - deviation to production system
    - mass-customized products and the requirement for the production system
- The ability to configure in marketing and production

The above issues are not trivial. This is mainly because without the visualization of configuration knowledge, it would be very difficult and time consuming, to even think the redesign of the platform. Thus, different approaches presented in the above list of benefits became reachable and well defined for the case company. As shown in the following sections, the configuration knowledge related to configurable products is very essential to understand in all levels of the organization in order to accomplish radical improvements for related processes.

4.2.2. Base machine estimations to confirm platforms

Using the matrix representation the base machines for all the product families could be defined by comparing the standard part with the variable part of the matrix. As the entire modular system is considered, the platform can be considered with respect to the entire modular system, i.e. the base machine modules or parts and components are compared with the variable modules or parts and components. Another possibility is to consider the delivered machines and figure out the base machine by using the customer based configured product structures to be compared with each other for the same product family. This approach was used to validate the presented platforms by the configuration experts. As expected, not all the information gathered from the experts' minds was accurate at the first time.

Nummela (2003) used the same configuration matrices for harvester heads and ERP databases to confirm the base machine presented by the matrices. The base machine presentation using ERP databases and sold customer variants over the life cycle of one modular system is presented in Figures 48 and 49. Figures 48 and 49 have been defined by comparing delivered customer specific product individuals and their module structures. The estimation has been conducted with harvester head H53 and harvester basic machine Ergo.

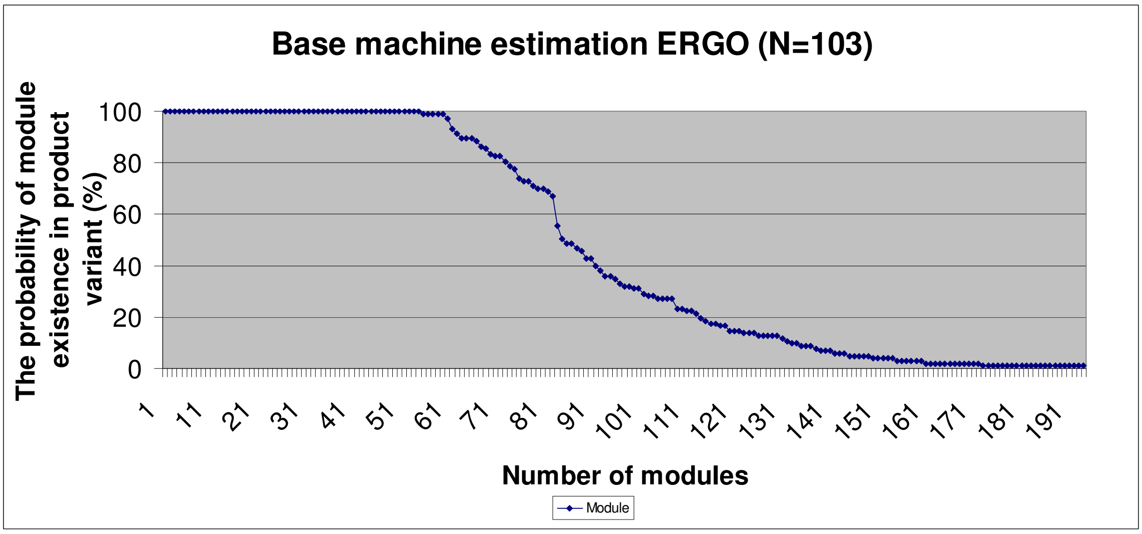


Figure 48. Base machine estimation using ERP databases for Ergo

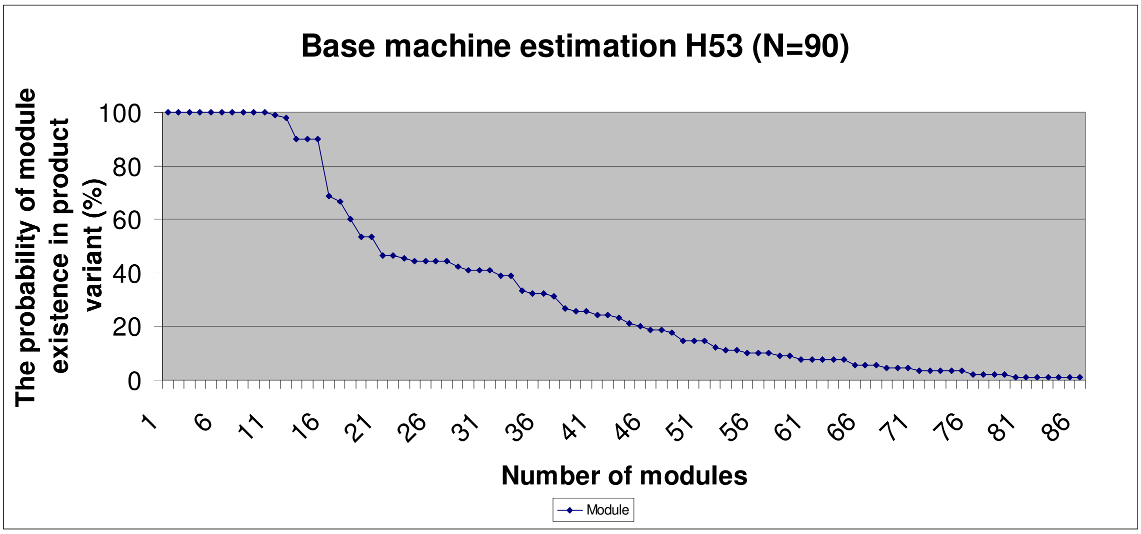


Figure 49. Base machine estimation using ERP databases for H53 (Nummela 2003)

As presented in Figures 48 and 49, the base machine is formed when the frequency of module existence equals the number of product variants produced during a time period (Nummela 2003). In order to reveal the base machine and in order to make sure that the base machine presented by the matrix is valid, it was needed to perform base machine analysis and iteration with the configuration experts. Figures 48 and 49 also show the variable part of the product during its life cycle next to the standard part of the product structure. Considering Figures 48 and 49, the base machine portion differs greatly between the two modular systems. For Ergo there are 64 standard modules where as H53 has only 10 standard modules. Thus, the intensity of configurability of the product structure affects the platform portion of the structure. The results of base machine estimations for harvester heads and their relation to matrix representations are shown in Table 8.

Table 8. Base machine estimations within a modular system (Nummela 2003)

Product	Estimation (Base machine %)	Matrix (Base machine %)	Average quantity of modules
H53	34.1	47.7	29.3
H73/H73e	16.9	33.7	29.6
H60	16.8	26.7	29.9
H60E/H60BW	17.6	25.5	34.2
HW60	41.4	41.4	31.4

The average quantity of modules is a long time average of produced product variants of a certain modular system. Base machine estimation uses the number of base machine modules in customer variants produced in relation to the average quantity of modules and the matrix-based base machine uses the same average quantity of modules and the base machine defined by configuration matrices. The differences between the figures are related to invalid configurations and problems with product structures, i.e. the matrix-based figure can be considered to be valid. While the above Table 8 considers base machines within modular systems, Table 9 considers platforms between the modular systems for harvester heads.

Table 9. Base machine estimation between product families (Nummela 2003)

Product	Estimation (Base machine %)	Average quantity of modules
All Harvester heads	14.8	94.6
H60/ H60E/H60BW/HW60	68.9	101.7
H73/H73E vs H60/ H60E/H60BW/HW60	24.8	101.0
H53 vs H60/ H60E/H60BW/HW60	21.4	93.5
H73/H73E vs H53	17.9	84.0

The average number of modules in this case is the average number of modules in the configuration matrix, i.e. in a modular system. The base machine percentage presents the modules standard to products selected into the analysis.

The above discussed analyses were considering mainly the harvester heads. The analyses based on ERP databases to figure out the platform modules include all the configuration mistakes and other problems related to the configuration process. In order to validate the configuration matrices, ERP based analyses were needed for every product being considered. Because for the case company all the products were described with configuration matrices and lots of iteration to form the matrices was done, the real percentages of the platform structure through configuration matrices are presented in Figure 50.

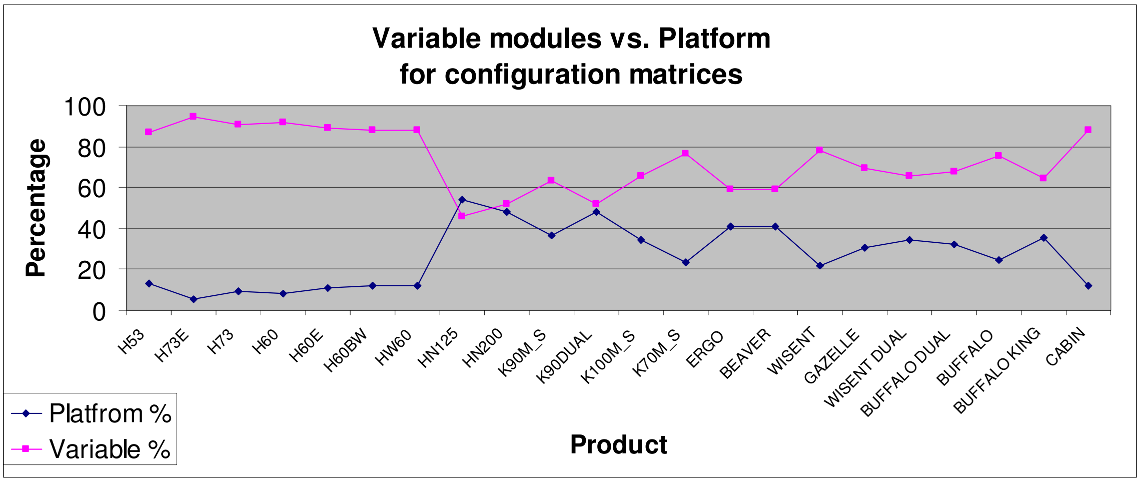


Figure 50. Variable proportion vs. platform proportion for the case company products

In Figure 50, the first 7 products are presenting harvester heads. Figure 50 points out that the harvester heads are very configurable while all the products have variable proportion of modules to equal over 80 percent. Also the cabin has variable modules for more than 80 percent of the total number of modules. Only one product has over 50 percent platform proportion of the product structure (HN125) in the modular systems analyzed. The percentages in Figure 50 are derived by analyzing the configuration matrices. Thus, the entire modular system is considered. This means that when a product is sold and configured, the number of platform modules stays the same while the configurable proportion is reduced by the selections made by the customer. When considering the long time average of modules in configured product structures, the following Figure 51 can be presented.

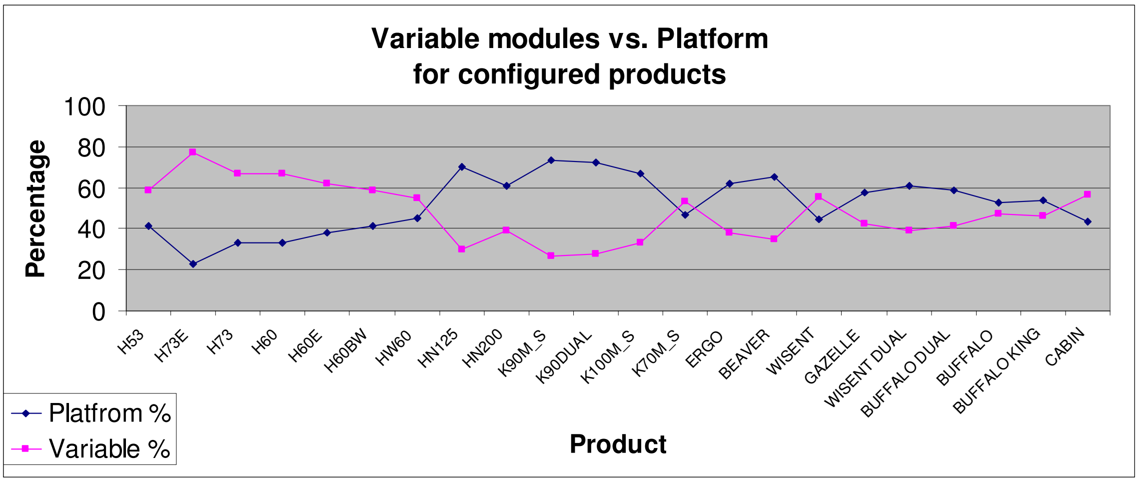


Figure 51. The platforms proportions for configured products

Figure 51 shows that harvester heads still have a larger number of variable modules than platform modules in configured product individuals, i.e. harvester heads are heavily configurable. The rest of the products in general have a larger platform proportion. For the company this is a good indication of the platforms available. As mentioned before the platform proportion is equal to the part of the product structure that can be freely decomposed into modules. For harvester heads this opportunity is very limited because of the variable nature of the product structure. Thus, heavy re-engineering of the platforms would be needed. For the other products the platform proportion is fairly large in size, i.e. the benefits for the production system could be impressive. Also, it needs to be noted that for cranes and harvester machines the number of parts and components in the modules can provide even better percentage for the platforms. This implies that the modules are very large in size, i.e. they are designed to meet the requirements of the configurability while the production system has been overlooked. Thus, the modules are too large in size and they could be decomposed into sub-modules or modules.

To consider these types of analyses the configuration knowledge needs to be well documented and in this case configuration matrices were used. The analyses made to reveal platform as well as modular system related base modules have been programmed so that they can be used at any time to perform new analyses the same way requiring very little effort.

In the case company nobody really knew that they even had platforms. By revealing the configuration knowledge and the base machines, gave a good understanding about the product structure and revealed the platforms. By doing this, the inputs were clear to the product design department enabling them to reconsider the production system more closely. Next section will consider the impacts of using the configuration matrices in the case company.

4.3. Results experience by the case company of using the configuration matrices

In this section the results experienced by the case company will be discussed. In sub-section 4.3.1 the tools based on the presented framework are discussed. Sub-section 4.3.2 considers the product structure evolution and the implications to the production system while sub-section 4.3.3 is concentrating on the implications experienced by the product development process. An example of the usability of the configuration matrices is discussed in sub-section 4.3.4 while sub-sections 4.3.5 and 4.3.6 are shifting the attention to the problems encountered during the implementation of the automatic configuration process and to the implications experienced in the configuration and configuration maintenance processes. Finally the feedback from the case company related to this research is discussed in sub-section 4.3.7.

4.3.1. Types of tools based on configuration matrices

In this sub-section the main tools based on configuration matrices for Ponsse are illustrated. The tools presented here were first illustrated in section 3.2 when the framework for configuration matrices related to the organization was established. Figures 20 and 21 present the tools that were developed on the basis of configuration knowledge. The main emphasis for the case company has been the development of production based tools to aid interpreting the level of modularity as well as to aid production development efforts. Table 10 presents the tools and their possible usage in the case company.

Table 10. Available tools based on configuration matrices

	Marketing and sales	Product development	Production
Configurator	x	x	x
Pricing	x	x	x
Change processes	x	x	x
Validity checks	x		x
Price inflation	x		x
Option frequency	x		x
Through put time analysis	x	x	x
Platform analysis		x	x
Commonality analysis		x	x
Cell re-engineering			x
Active module based analysis			x
Invalid parts analysis			x
Product kill analysis		x	x

In Table 10, purchasing is considered to be under production. As seen from Table 10, the emphasis has been to satisfy the needs of the production department, thus all the tools are applicable to production. One of the main reasons that the production needs these types of tools is that the queries related to issues presented in Table 10 are considered the responsibility of the production department for the case company. This implies that a serious lack of co-operation between departments is the reality for the case company. Even though the development of the tools presented in Table 10 started from production, the result was that when holistically considering issues, such as modularity and configurability, the entire organization could be involved more easily. Thus, many of the tools presented were also useful for other departments and processes throughout the company. Appendix 3 further presents the tools illustrated in Table 10.

4.3.2. Product structure evolution using matrix representation and implications to production system

As mentioned before, all the product structures analyzed during this case were based on functional modularity. Considering the evolution model provided by Lehtonen et al. (2003), functional modularity is meant to support configuration, product development and sales. For Ponsse this approach has been selected mainly to satisfy the needs for sales to be able to sell options and for production to define the customer based product structure as easily as possible. This type of modular system has proven to be one of the competing edges for Ponsse next to the marvelous products. In the beginning of this research the problem was that the current situation and level of modularity was merely tacit knowledge mainly in the minds of the production planning personnel.

In the case of functional modularity, saleable options correspond fairly well to the modules. The result is an easy way to build configuration matrices and to configure products. The tediousness of building the matrices as well as understanding the current modular system is related to the combinations that are formed between the options. When all the matrices were formed, analyzing the product structures actually revealed the next level of module evolution from functional modularity to the direction of customer oriented platform based modularity presented by Lehtonen et al. (2003). In Figure 52 one entire base machine is presented for harvester machine ERGO. Figure 52 is only a part of an entire configuration matrix.

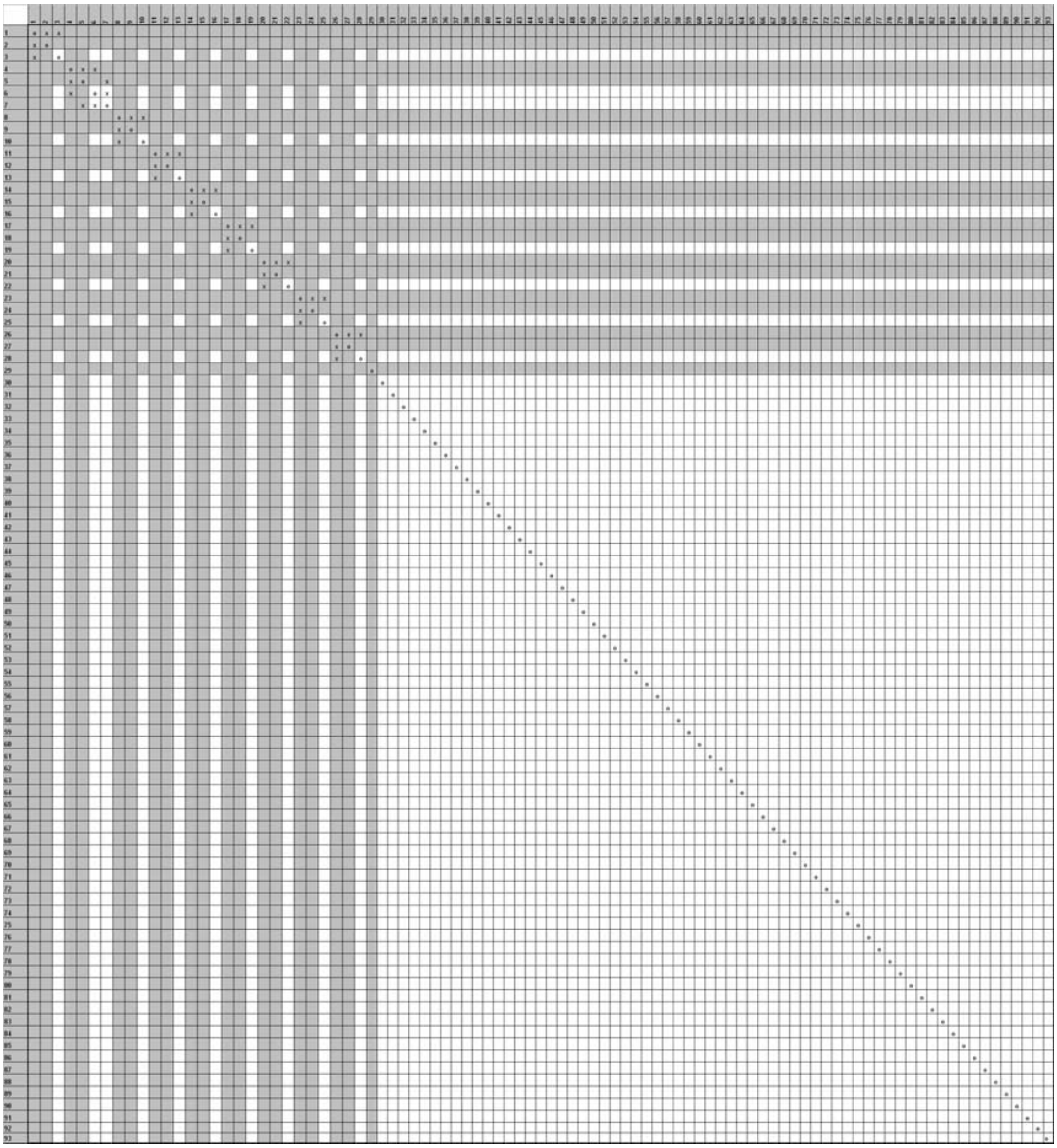


Figure 52. Base machine

The total number of modules for the entire modular system for the case product totals 184 modules. From these modules, 64 (Figure 52 bottom) have no dependencies between any options, i.e. these 64 modules form the platform for this modular system. When selling one variant of the presented modular system, the average number of modules equals 100, i.e. considering the number of modules the base machine covers 64% of the entire product structure. The reason why the base machine is so important is that it is always standard for every sold product variant from the modular system it represents. For the case study this part of the product was also functionally decomposed. Considering Figure 14 the implementation of platform-based modularity to reduce the cost of customer variation is to figure out the standard part as well as the variant part of the product structure. The base machine does just that. To consider this issue more deeply we will now consider the impact of functional modularity to the production system as well as the impact of the base machine approach to the production system.

When functional modularity is used as the main type of modularity for the entire organization, the final assembly structures suffer the most. None of the functional structures correspond to the needs of the production. This issue is clarified in Figure 53.

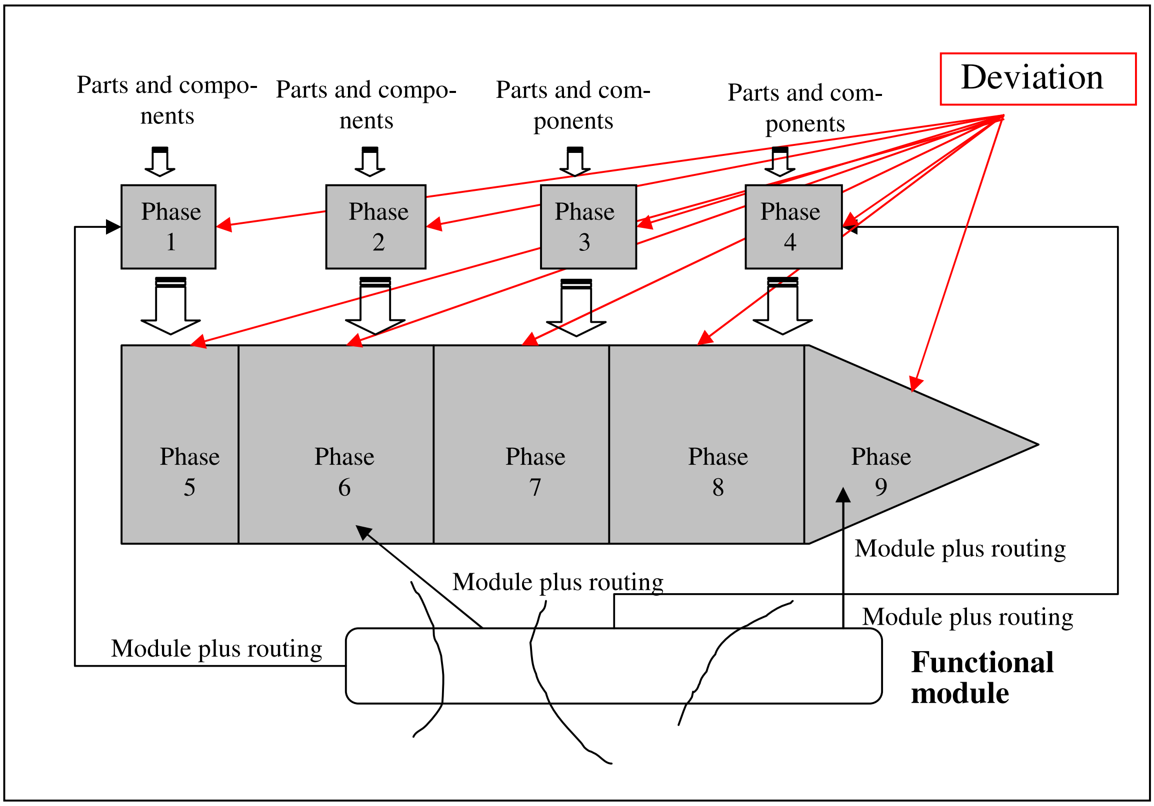


Figure 53. The decomposition of functional module into the production system

Figure 53 presents the decomposition of a functional module into the production system including preassembly cells and an assembly line. The entire customer specific product is a list of functional modules in the system. Considering the nature of a functional module, routing information simultaneously with the module number identifies needed material for different phases of production. The entities of material are not necessarily subassemblies, but usually they appear as a group of parts and components needed in a certain phase. While the entire product is decomposed in such a way, the following problems arise:

- No subassemblies or assemblies defined for production
- No standard time system is easily possible
- Capacity planning is hard
- No labor in the ERP system
- No drawings for production

- Outsourcing is impossible without iteration between production, purchasing and product development

The problems from functional modularity are severe, while the company can still manage with this type of product architecture. How well the company can handle functional modularity is related to its outsourcing and production strategies.

The deviation in Figure 53 is meant to show that in the case of functional modularity the production system experiences deviation virtually in all the cells in the production system. This is due to the fact that variation for product structures cannot be isolated into some part of the production system since it is not well known, i.e. routing information can be considered tacit in many ways. The result is that usually the personnel building the products know the products so well that they do not use work order instructions, but only consider sold options to be able to vary the final product. To reduce this deviation the product structure needs to be defined by other means than only by functions.

As presented by Lehtonen et al. (2003), the stage after functional modularity is concentrating on cost-effective customer variation. They show that the product structure needs to be divided into standard part and varied part. The base machine presented in Figure 52 shows the possibilities to transform the product structures into standard and varied parts. The conclusion for Ponsse's products is that decomposing the standard part of the product structure into functions is over-modularization, i.e. there is no point in using functional modularity for the standard part of a modular system. This means that the standard part of the product structure can be decomposed considering the production system to reduce costs and deviation. Figure 54 shows the base machine approach to satisfy the needs of cost-effective variation.

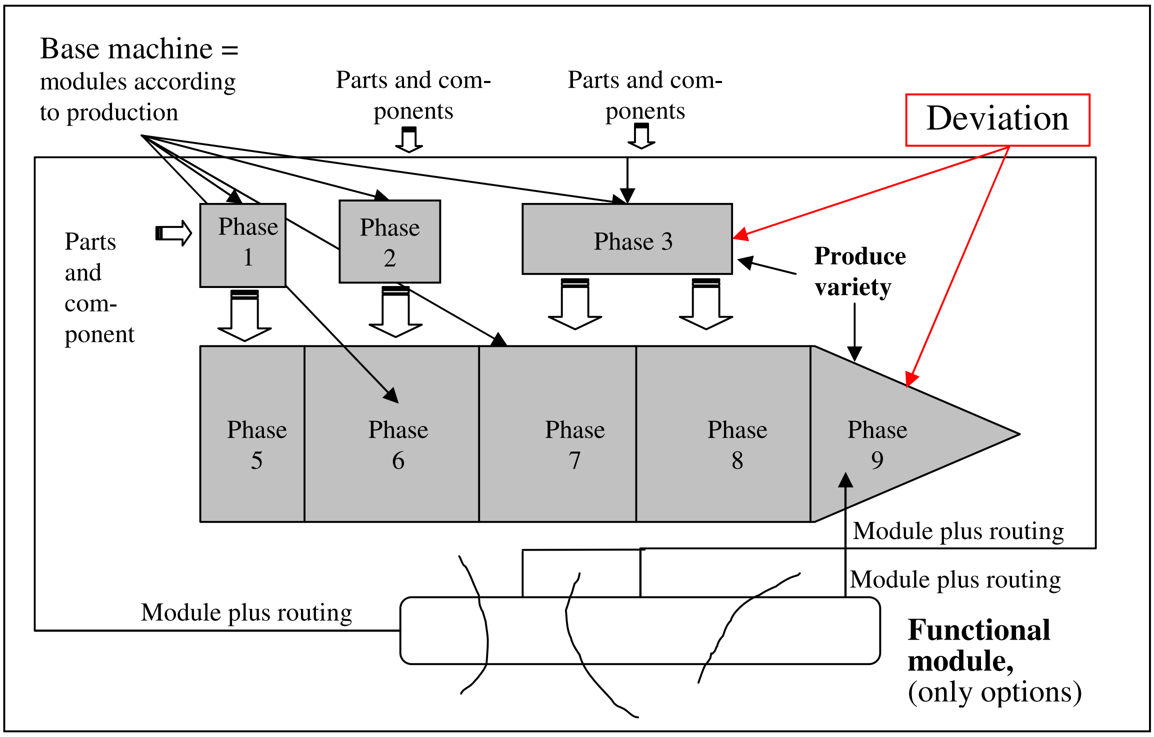


Figure 54. The base machine approach

The main idea of decomposing the base machine into standard modules is to provide production with means to handle and isolate deviation to certain parts of production. The base machine approach means that the standard part of a modular system is decomposed considering manufacturing and the customer variation is handled with functional modules. It is noticeable that the functional modules should be designed as large as possible (configurability is the limit, consider MBI). If functional modules are large enough, there is a possibility to decompose the function into standard subassemblies and parts and components, i.e. the negative effects of functional modularity can be minimized in the context of production.

Considering the configuration matrices that have made this approach possible for Ponsse, the structure of the matrices is developing into the direction of fewer modules in the standard part while the number of parts in these standard modules increases. Also the module interfaces can be re-evaluated since the standard part can be located into the base machine modules.

The main issue for Ponsse was the realization of the current situation of product modularity through configuration matrices. This idea of base machine approach goes hand in hand with the concept of producibility presented in section 2.5. One of the main ideas with producibility is the similarity between the production system and product structure simultaneously with a configurable product structure. For the case company the shift from functional modularity systematically into customer-oriented cost-effective platform means more producible product structures as well as:

- Possibility to use standard time systems for standard parts of the product
- To include standard times into the ERP system
- To handle and isolate variation in production
- To use delayed differentiation
- To understand modularity holistically
- To estimate capacity needs more carefully
- To have the possibility to decide between MRP and other marshalling methods
- To build possibilities to respond to sudden capacity needs by shifting assemblies to and from own production
- To systematically take the needs of production into consideration while designing the product

Delayed differentiation (see e.g. Selladurai (2004) and Partanen and Haapasalo (2004)) provides means to use pull and push system simultaneously. The line should be pushed while the rest of the production should be pull-oriented. Thus, the ideas of mass customization will be satisfied and faster deliveries could be achieved. Modularity enables many important features while it needs to be designed into the products holistically. By further understanding modularity and developing product design processes, the company can improve its capabilities of inducing modularity into the product structures. By doing this there is a possibility that product design can provide more and more producible products while simultaneously configurability is well under control. The production system as well as the evolution model has been covered in this section and the following sections are concentrating on the implications of the configuration matrices and base machine approach into the product development process and configuration process.

4.3.3. Implications to product development process

As noted above, the evolution of modularity can clearly be seen by using the configuration matrices. As stated many times, modularity should not be considered by only one aspect, i.e. modularity should be seen broadly as means to serve the needs of an organization. As producibility is needed also from product structure and its modularity, it needs to be designed into the product. The processes and organization of product development are affected.

When designing new products, configuration matrices should be the responsibility of product design and after the product has been launched into the production, the maintenance of configuration matrices should be in the production related organization or even in product design department. The latter case would provide the most efficient way of maintaining configuration knowledge for the entire organization. The new product development process chosen to be used in this thesis was heavily based on configurability of the product structure as presented by Aarnio (2003). In this process as well as in the other product development processes presented in this thesis, the product structure and decomposition of product structure into modules was one of the first steps considering the product development process. Thus, modularity should be designed into product structures while simultaneously taking all the stakeholders into consideration.

The case company had problems relating to prototyping and production ramp-up which are partly due to the late formation of product structures and modularity during the product development process (see Figure 46). As modularity is not purposely designed into product structures by product design, the iteration at the end of the product development process between production and documentation is noticeable. If the modular product structures should be defined by using the ideas in Figure 24, the organization and processes would be heavily affected.

As the level of understanding about modular product structures increases, the aim for product development is to deliver producible product structures for the rest of the organization to work with. The problem for Ponsse is that product structures and modularity are seen as production issue which is mainly due to the model for generating modular structures (Figure 46). The biggest mystery is that while a single designer is responsible for a certain entity of a product, i.e. module, the modularity designed by the designer is lost when documentation takes over and produces drawings of their own to be delivered to production for module creation. At this point the risk of iteration is probable since production and purchasing can have their requirements which are usually taken into consideration at this point, i.e. lots of module interfaces can be affected if drawings are reorganized.

Changing the environment of product design in order to address issues relating to configurable products affects the entire product design organization as well as processes involved. Considering the case, the formation of product structures needs to be turned around and the modularity of the product structure will become one of the drivers to deliver producible product structures. The pressure for the case company's product development process is shifted to the inputs of the product development process, i.e. the pressure for product planning to define proper specifications to product design is central. Issues such as customer needs, commonality plans, variety plans and technology decisions need to be addressed and defined, i.e. as production needs solid inputs from product design, product design needs solid inputs from product planning to secure systematic processes.

The MBI method (Aarnio 2003) has been selected for the case due to the aspect of configurability and the early formation of modular product structures. For Ponsse the use of MBI is still far away due to the process of understanding the current situation of modularity for many people is still under way. During this study there were no possibilities to try out the MBI method for new product development. The main benefit has been the fact that the inputs for product development have now been clearly defined and the strategic process for determining these issues is under way in the case company.

4.3.4. Disintegrating the cabin structure with configuration matrices

A project related to product design was carried out during this thesis. The goal of the project was to disintegrate the cabin structure from the basic machine. The basic machine means the part of the final product that is assembled in the line. The triggering issue for the disintegration was the simultaneous project to re-route the assembly operations. Re-routing needed the disintegration because the cycle time for the cabin was too big to handle while integrated with the basic machine.

When considering the cabin structure, there is no sense in having this big module integrated with a part of a machine that is produced in a different location. Furthermore, this part of the product is heavily configurable. In configuration matrices this means that the features selected to the cabin were integrated into the basic machine matrices making these matrices more complex. To make this project more challenging the time that was allocated was very short (two months) and the fact that this had been tried and failed before made it even more motivating.

To get an idea of the problem, the existing configuration matrices were carefully studied. By the use of configuration matrices, drawings and ERP databases the following critical factors were defined:

- All the active modules
- All the configuration rules
- All active features and their options
- All the active drawings
- All the active module BOMs
- All the phases related to parts and components in BOMs

The problem in the drawings and BOMs was the fact that there were simultaneously parts and components from the cabin as well as from the basic machine. To make it even harder the seat and the electrical works were included into the analysis.

To make it all happen there were two additional matrices defined on the basis of the active modules. These configuration matrices were used to get an idea of the entire cabin structure, split into harvesters and forwarders. By doing this the preliminary modules and features were defined. The formation of these matrices was done by using the information gathered from the available configuration matrices as presented above. By defining the modules that were related to cabin and by defining all the features that related to the modules, the additional matrices were defined. When considering this case, also the basic machine matrices and the module level were affected making this phase very critical. The quality of information when finalizing all the related matrices is of great importance. Without these matrices the disintegration would be very tricky and could be even impossible.

When having the drawings and the configuration knowledge easily available there were no problems when re-engineering the module interfaces. This was the key enabler for the entire project. When re-engineering the interfaces the requirements for the old modules as well as for the new modules were very easily defined using the configuration matrices. Also one of the main benefits of the configuration matrices was the fact that many options needed to be selected could be eliminated through a careful re-engineering, i.e. the configuration matrices were made simpler.

The problem was that the cabin was very configurable producing one of the most complex matrices. After the preliminary matrices had been used for disintegration purposes the two preliminary matrices were integrated to form a new configuration matrix for the cabin. By doing this the team could locate the re-engineered modules into the final matrix, i.e. the new configuration matrix for the cabin was established. For the cabin structure, the results are presented in Table 11.

Table 11. The results from the cabin case

	Modules	Part rows	Different parts	Drawings	Platform modules	Platform parts
Old situation	116	2062	565	162	5	136
New situation	98	379	258	82	12	249
Change %	**-15.52**	**-81.62**	**-54.34**	**-49.38**	**58.33**	**45.38**

The results are taken from the preliminary configuration matrices versus the final matrix. It needs to be noted that not many parts or components were eliminated since no re-engineering (only interfaces were re-engineered) was done. Parts and components were only shifted between modules and configuration matrices. What is important though is the enlargement of platform modules and platform parts 58.33% and 45.38% respectively. Also the reduction in drawings (49.38%) is significant. All of the results in Table 11 mean that the cabin structures and their configurability are more easily handled and the information related to cabin drawings are accurate. Also one new configurable entity was introduced to be configured during the order delivery process and by doing this the configuration models of the basic machine were simplified greatly.

This project was a subproject for the re-routing project. The success of the disintegration project enabled the success of the re-routing project. The final outcome was that the disintegration was done in one month by the researcher with one aiding resource supplemented by random help from organization. Without the configuration matrices even the definition of active modules would be difficult to accomplish, not to mention the work that would have to be done to get an idea of the configurability of the structure. Configuration matrices made the complex project easily executable.

During this process new configuration matrices were introduced and for the production planning personnel there were no experiences of the new configurable entity created. The long order book needed the reconfiguration of tens of machines (both the cabin and the basic machine) and the first steps to the configurator and automatic pricing system were taken. Due to this project the courage for a self-made configurator was gathered and the configurator presented in section 3.8 built.

4.3.5. Problems encountered when implementing automatic configuration process

For the case company, documenting the configuration knowledge was one of the main issues to be dealt with in order to secure the competitive edge related to customer service during the order-delivery process. For configurable products, configuration knowledge represents the core information related to product configurability and it enables the use of systematic configuration processes, both manual and automated. As discussed above the importance of configuration knowledge can be seen next to the configuration process company-wide, as seen also in the case of Ponsse.

Configuration knowledge for Ponsse has traditionally been "documented" in the heads of production planning personnel and the role of these persons has been central for the effective manual configuration process. To make this role even more important, these persons are also responsible for generating modules to be used when configuring and producing the product (Figure 46). As marketing and sales determines the list of options to be sold fairly independently, there are usually conflicts of what production can deliver and what marketing thinks production can deliver. Simultaneously with these issues, time pressure for product design and the flexibility of the case company processes cause incomplete designs in the context of configuring the product. This means that product development knows that some options will be sold in the future, but in order to respond to time-to-market pressures these options will be designed after the option has been sold by marketing and sales. Thus, marketing and sales has the possibility to sell properties that are not designed yet. This situation offers maximal flexibility to the customer, but the problem is that marketing and sales sell whatever they need to close a deal, product development has no time to deliver complete designs and production works with incomplete knowledge between product design and marketing and sales.

The problems presented above are mainly due to the lack of systematic processes as well as to the lack of understanding the configurable products and modularity in a broad sense. Configuration matrices solve these problems partially by giving the understanding and clear inputs for different parts of the organization. Consider, for example, the decomposition of the standard part of the configuration matrices. This part will enable the product design to provide the basis for platform thinking and for cost-effective production in the context of configurable products. Configuration matrices also define exactly the features and their options to be sold. This will eliminate differences between departments since the configuration knowledge in the configuration models is valid. The differences between the departments of the organization tend to cumulate into the configuration process and inconsistencies here will only provide deviation to other parts of organization.

Problems still remain while establishing an automatic configuration process. Changes affecting the configuration knowledge and process were presented in section 3.7. The presented changes hold true for the case company and the problem was to handle this turbulent environment. Even if it can be stated that processes can be re-engineered in order to eliminate these changes, there will always be changes to the product designs during the life cycle of the products as well as changes to the customer specification during the order-delivery process. This is the reason why the processes in section 3.8 were defined. After the cabin case there were the following possibilities to automate the configuration process for the case company:

- Configurator supplied by the PDM system
- Configurator supplied by the ERP system
- Third party supplied configurator
- In-house developed configurator

The first three possibilities were closely studied and while the current ERP system used in the case company included a configurator it was tested profoundly. It soon became apparent that when a configurator was built by heavily considering the rules generator, it was easy to setup the configurator by using the configuration matrices but the problem was that when changes took place the updating work became very tedious. This was due to the fact that none of the three first configurators provide automatic updating of the configuration rules. This really meant that even if the feature structure could be changed, changing the configuration of a configured product became almost impossible and very time consuming. It also required a very deep understanding of the configurator. Also when the configuration knowledge was heavily changed, the manual work needed to update the configurator became more than difficult. This was due to the fact that it needed to be done manually by observing the configuration rules from the configurator and comparing these rules module by module to the knowledge provided by the configuration matrices. This also meant that reconfiguration of orders in the order book became impossible. Mainly because of the issues presented above, the first three configurators working mostly with similar principles became too problematic to implement.

Even if the configuration matrices are in use, the amount of maintenance related to configuration knowledge became one of the main obstacles for the case company to enable an automatic configuration process. As shown in Figure 29, both product design and marketing affect the configuration process and the knowledge related. The problem is the version changes of modules from product design affecting the configuration knowledge presented in configuration matrices. These changes require the change of configuration models, both for the documented configuration matrix knowledge and configurator system knowledge. For Ponsse, the average rate of change is 2 module changes per day which implies that there would be large amounts of configuration model versions and updating work to be done. Even if configuration matrices can be integrated into the configurator and the configuration models can be automatically uploaded, the usage of a configurator with the current processes can be questioned. This is the stage where process re-engineering is required since systematically handled configurable products put enormous pressure on product development and marketing to deliver suitable outputs. For Ponsse, these requirements are clear due to the configuration matrices and to the careful analysis conducted. The main requirements in order to make the automatic configuration process possible are:

- Deliver systematically modular and configurable product structures for the rest of the organization
- Systematize processes in the context of configuration knowledge and know the effects of your own actions in respect to other stakeholders of the company
- Concentrate on change processes to enable systematic and well-defined processes to work with

For this case, it is apparent that configurable products need systematized configuration knowledge and suitable organization to work properly. For the case company the configuration knowledge is systematized. But the automatic configuration process places more requirements on configuration knowledge and especially to the processes related to configurable products. This requires broad understanding of configurability and modularity by the organization as well as re-engineering of many processes to systematically address the problem of configuration.

Next to the change management, the issue to solve was the storing location of configured product individuals. First of all, the product development process determines the product structures to be used and enables the company to handle the products' life cycles. Using the ERP system to configure the product structures is adequate for the order-delivery process, but after sales can be affected if the full product structures are not maintained in ERP. While using PDM to configure products the after sales would be secured and in order to make the production possible the integration between ERP and PDM system is needed to deliver product structures to be produced. Using a third party supplied configurator integrates the systems, as the configuration knowledge from configuration matrices in the case company would be uploaded into the configurator and the configured product structures into the ERP and PDM systems for maintenance (after sales, PDM) and production (production, ERP). A third party supplied configurator would be the most flexible in this sense, if the integration between systems can be established.

The above discussion of the problems related to the first three configurators lays the ground for the fourth one. The above problems, the features of configuration matrices and the cabin case provided enough understanding to establish an in-house built configurator. This configurator is deeply discussed in section 3.8 and the benefits are considered in the following sub-section while appendix 4 compares different configurators.

4.3.6. Implications to the configuration and configuration knowledge maintenance processes

For the case company the process for making the automated configuration of products possible started by using the configuration matrices for the manual configuration process first. As configuration matrices were large in size, more sophisticated methods were needed to help studying the matrices. In the first phases manually configured products were also double-checked by another person after the configuration task was complete to check the validity of the configurations. The configuration process was defined by establishing an accurate description of the entire process (example in Figure 55). The amount of steps for the old manual configuration process for the first level is shown in Figure 55.

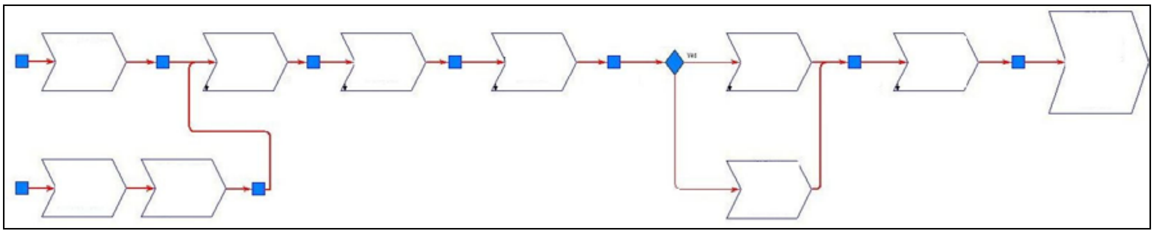


Figure 55. The old configuration process (rough sketch)

Figure 55 is only provided to see the amount of steps in the first level of the configuration process. Thus the complexity of the process can be evaluated. For the configuration process that starts with receiving an order has two parallel processes, i.e. a process for domestic and export products. There is no reason for this since all the products use the same configuration models, thus only one process is needed. After the steps in Figure 55 the order is sent to the production planning personnel that takes care of the configuration task for the product.

It can be noted that the processes include tasks that define if the customer specification is valid or not. If not valid, the order center will be contacted for further information and they can contact the customer if serious configuration problems exist. In reality the process for receiving the order up to the point where the product individual is actually configured involves many people from many parts of the organization and a complex process with lots of useless data interchange between people. As mentioned in the beginning of this chapter, the wasted time related to the above process was over 90%. The process actually includes this waste systematically since the configuration task is considered to be done one month prior to the production start date.

For the new process the framework provided by the configuration matrices influenced the processes deeply. First of all, the configuration matrices provided very systematical ways of maintaining the configuration knowledge and also a tool to give the exact information for module interface and dependency maintenance. The configuration process was also greatly improved by the use of a configurator. The configurator built is totally dependent on the configuration matrices which provide all the information needed to perform the configuration task. Also the task of updating the configurator is eliminated by the use of the configurator rules generator that provides configuration knowledge and a new revision automatically. Thus, all the configuration models are automatically converted into databases for the configurator. By revisioning the configuration knowledge into the databases provides the basis for efficient change management. Even if the processes would be ideal for configuring products, changes still take place during the life cycle of the product as well as during the order-delivery process. The case company can now use the configuration knowledge very efficiently to make sure that all the changes are made with the latest configuration knowledge. One of the biggest issues is that the case company can now also handle the revision changes next to the changes related to module versions. This means that the order books will include only valid information related to configured products. This has not been the case previously. The new process steps developed are presented in Figure 56.

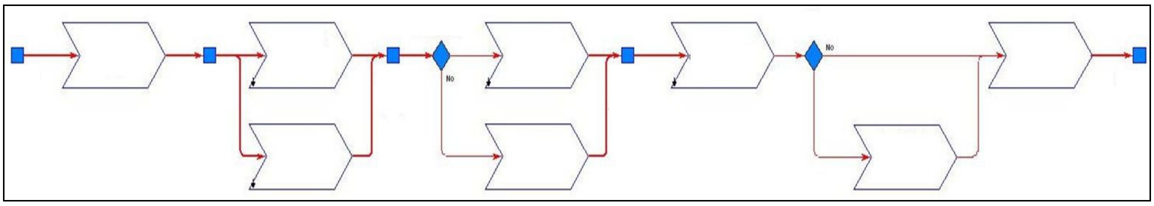


Figure 56. The steps for the new configuration process (rough sketch)

The process in Figure 56 includes all the steps needed to configure an order. The parallel process for export machines was eliminated and the e-mail process automated to provide the customer specification to the applicable stakeholders. When the information was readily available in the right form at the right time, tasks could be made simultaneously in different parts of the organization, i.e. the configuration task needs not to wait for the other tasks to be completed. The process using the established configurator is as follows (the process in Figure 56):

- Receive incoming order
- Setup a new sales order in ERP
- Setup a new PDM project for the order (use the sales order number from ERP)
- Configure the customer specific product individual
- Update the forecast

The steps are very fast to execute and if there is integration between the configurator and ERP or PDM systems the configured product structure can be uploaded to the selected system automatically. The new process provides the possibility that when the order is configured, all the order information in all the appropriate systems is valid right after the configuration task is completed. The lead time for the new process equals the time that it takes to configure the order with the configurator. The tasks that have been made parallel (usually the needed information to the sales order) are done by clerks not related to the configuration process. The task of updating the forecast was shifted to the production planning personnel from the order entry department. This was mainly done to streamline the process and to reduce the workload from the order entry department.

As mentioned before, there is no reason for this configurator to disintegrate the production and marketing configurator. When the first steps were taken to implement the configurator, these two configurators were separated. The idea was that when marketing configures the customer order, the information is sent to the production planning personnel that use the production configurator to establish the product individual module structure. This way all the inconsistencies from the configuration matrices were removed and the quality of the models improved. This was also very important for the production planning personnel so they could understand the entire configuration process from start to finish. A very effective configuration team was formed during this period.

Previously the customer specific changes were very easily accepted and done without too much consideration. For the new process the changes need to be established with the configurator and it is the responsibility of the order entry department to take care of the changes. Only the change information is sent to the production planning personnel and other stakeholders so that they can react to the new situation. As mentioned before, the marketing and sales could have different ideas of saleable features than the production planning personnel. Due to the configuration matrices, the misunderstandings are eliminated since when using a configurator, sales cannot really sell anything not presented in the configuration models. Through the use of the models marketing and sales are also aware of all the changes to the products made by product design. This shows that the co-operation for the case company between production, marketing and sales and product design has been improved as well. Thus, tacit knowledge has been decreased related to the company's configurable products and processes related.

After implementing the configurator, the work of the order entry and configuration personnel shifted into direction of less routine and monotonous work. For example, the people used to configure the products will have more responsibility of maintaining the configuration matrices and react only to change situations when needed. Also the reconfiguration of the customer orders when configuration models change is their responsibility. Thus, if the new configuration knowledge has changed so much that validity algorithms will not be passed these people will reconfigure these orders, i.e. the order book will only contain valid configurations using the latest configuration knowledge.

After all the elements were in place to enable automatic configuration process, the benefits could be seen clearly. Table 12 presents the situation before and after the configuration matrices in the context of automated configuration process.

Table 12. The situation before and after configuration matrices

	Old situation	New situation
Configuration task in sales	manual (errors likely to happen)	web-based configurator with valid configuration knowledge
Configuration task in production	manual (errors likely to happen)	automatic with valid configuration knowledge
Information from orders	e-mail sent manually (includes the contract done with the customer)	automatic e-mails with a valid configuration specification (no work related)
Configuration models	ill-defined	well defined
Configuration knowledge maintenance	ill-defined	well defined
Configurator database update	not applicable	automatic with revision knowledge
Double checking the configurations	critically needed	not needed
Revision changes	not updated	automatically updated
Version changes	updated manually	updated automatically with well defined algorithms and processes
Change interval from product design	continuous	well defined intervals (critical changes only established immediately)
Order book configurations	not updated	updated automatically with well defined algorithms and processes
Customer changes	reacted with complex processes	reacted with well defined algorithms and processes
Information from changes	manually done in many steps	e-mails automatically to right people with the right information
Reconfiguration	manual with tacit knowledge	easily done with a configurator and revisioned databases
Pricing	basis not well established	basis automatically established
Learning configuration models	months	minutes (only basis for configuration matrices needed)
Knowledge dependent on personel only	heavily dependent	not dependent at all
Wasted time related to configuration process	over 90%	0%

The comparison in Table 12 is divided into configuration process, configuration models and their maintenance, customer based changes, and finally to pricing, learning, personnel requirement issues and wasted time. As seen from the above table, the improvements for the case company processes have been tremendous. The most critical part of the configuration task and maintenance has been automated and processes well defined. This has enabled the case company to reconsider the resources needed to configure orders and maintain configuration knowledge. It has been mentioned that previously the aim for the company has been to put enough resources to the configuration knowledge and module maintenance since there are no other possibilities to secure these critical issues. The result was that currently this task can be handled with one person and when further developing the product development processes (e.g. the use of MBI) the workload of this person can be further diminished. Also when considering new employees, the complex configurable products can be easily learned through studying the configuration matrices and using a configurator for test use. Thus, only the idea of configuration matrices needs to be taught. As mentioned in section 4.1.4, the waiting time for the papers during the configuration process was over 90%. For the new configuration process the wasted time is virtually zero due to the fact that the configuration is ready right after the marketing configuration has been finished.

Even if the in-house built configurator provides the most efficient way for the case company, the emphasis should be on the three main requirements (see sub-section 4.3.5) to make the configuration process presented above automatic. While improving processes for product design, marketing/sales and for production, the configurator will work as planned while continuously providing accurate information due to the well defined processes and algorithms. Improving the processes will provide more efficient use of configuration knowledge while simultaneously providing more accurate and quicker processes for the company. These requirements are now well established for the case company.

The above discussion has also been a good indication of the critical nature of the configuration knowledge related to the entire organization while simultaneously configuration knowledge and modularity are heavily integrated. This makes the organization integrated as well and concurrent engineering is required in order to design producible products.

4.3.7. Feedback from the case company

During this study the role of the researcher was mainly to balance between production and sales, production and product development as well as between sales and product development. Configuration matrices were the glue between the different departments. Also the configuration maintenance personnel were working very closely with the researcher. The feedback presented in this section is collected by discussing with the personnel involved during the research.

Configuration knowledge maintenance concluded that without configuration matrices the maintenance of the modular products for the case company would be close to impossible. This is natural since there is only one person to perform this task for the entire spectrum of products. They were also very relieved that finally development efforts were also directed to aid their work. As this team is also responsible for the manual configuration task, they felt that after the automatic configuration process was established, the manual work related to transferring the product structures from the configurator to the PDM system became boring. There was only a need to tab module after module to the system. This was corrected after the integration between the systems was established. This part of the company was very committed to using configuration matrices. Even when new personnel were hired, the old staff would automatically teach the use of configuration matrices for the new employees. Commitment of these people could be seen as positive feedback from using and maintaining the matrices. This did not happen over night though. It needed the cabin disintegration because after that the production personnel did not understand the cabin structures, i.e. they needed to use the configuration matrices in order to make sense of the product structures. Thus, they understood the usability of these matrices.

For marketing, the operations manager responsible for receiving incoming orders was happily surprised when the configurator process implemented was working as planned. He also commented positively on the automatic pricing process that gave basis for complex cost estimations based on configuration knowledge.

The production development personnel often require analysis related to the tools presented in this chapter. As these analyses usually take under 5 minutes, the tools were frequently used and further developed. One of the main issues for these people was the generic product structures that were frequently used when designing new layouts for production. Also the support for the routing project was appreciated by the production development personnel. This project would have been stopped if the cabin disintegration had failed.

The plant director responsible for the production and product development sees the role of configuration matrices more strategic. According to him, the configuration matrices gave the case company the understanding of the integrated nature of the production and the product development. Also the understanding of the relation between modularity and configurability became clearer in the context of the production system as well as in the context of the order process. He also pointed out that the evolution of modularity and the meaning of product structure for the business in configurable products was becoming more understandable. Also the aspect of producibility of the product family structure was pointed out by him. He concluded that the new approach integrates the entire business in a way that modularity and configurability can be used effectively, thus ideas presented in this thesis helped the company to see how product structure integrated product development, manufacturing and assembly and after sales with the help of configuration matrices.

For product development the revelation of configuration knowledge was appreciated but at the same time it was considered to be the responsibility of the production planning personnel. Even if the inputs for changing the product development processes were very clear, no actions were taken. For the platform proportion of the configuration matrices, product development started a project to redesign the module interfaces. The problem was that three-dimensional models were needed to be accomplished before the redesign could be finished. The results of this preliminary project were promising. The reaction of the product development was surprising since the biggest benefits can be harvested through further developing the product structures as presented in this thesis. One of the main obstacles was the fact that product design considered further modularization to be the routing job that was also the responsibility of the production planning personnel. From this, the level of concurrent engineering can easily be concluded.

Even when the research ended the development and maintenance of the configuration matrices were located into the production planning department. Considering the rate of growth for the case company, the emphasis will be on platform structure in the future. Thus, also the organization of the product development needs to change. This is beyond the scope of this thesis. The following chapter will conclude the results discussed in chapter 4.

5. RESULTS

This chapter considers the main results and discusses the role of the configuration matrices in organization generally. The role of the configuration matrices is discussed in section 5.1. Section 5.2 considers producibility, modularity and configurability in the context of configuration matrices.

5.1. The role of configuration matrices

The aim for this research has been to focus on providing an easy method to visualize configuration knowledge. Another point of view is to use configuration knowledge as an integrative element between different stakeholders in the company. Thus, product structure ties different disciplines together through generic product structures. Sections 5.1.1…5.1.4 conclude the role and benefits of configuration matrices in organization.

5.1.1. The role of the matrices in product development

Product development processes have been discussed earlier in this work and the realization of the product structure has been concentrated on. There is also a question of the role of product development on creating configuration models. Product design is responsible for documenting the configuration model to be used in subsequent phases, i.e. as modularity is the responsibility of product design, the creation of configuration knowledge documentation is also the responsibility of product development. There are also other possibilities for creating configuration models by offering the responsibility to other functions. This action most likely increases iterative work in the organization as the knowledge is shifted and the aspect of configurability is moved away from the product development process. The main benefit for product development using configuration matrices is the systematic way of documenting configuration knowledge. When configuration matrices are integrated with modularization method such as MBI, the configuration knowledge can be documented during the product development process.

After the product development for a new product has been completed, the configuration matrices and the related knowledge are maintained by configuration maintenance. This process needs to be connected to the product development process that maintains the existing products in order to keep the models valid. Change management is one of the key elements. The configuration matrices can be used to document the modular product structures developed by product development. Configuration matrices can be created after the decomposition of the product structure and after the module interfaces have been defined. The use of configuration matrices is product family dependent, i.e. one configuration matrix is needed for every configurable entity.

While configuration matrices hold all the knowledge related to a product family, the matrices can be used to analyze the product family structures. The power of configuration matrices is related mainly to the generic product structure next to the establishment of configuration knowledge. For product development the following tools were constructed:

- Platform analysis between product families
- Inner product family platform analysis
- Commonality/differentiality of modules
- Commonality/differentiality of parts and components

Next to the analysis presented above, configuration matrices hold valuable information about the state of the modularity in different products. Inconsistencies in matrices concerning the saleable options and designed modules can be easily defined and corrective actions taken if needed. For product development the use of configuration matrices can also be seen as the integrative element between product development, production, marketing and after sales, i.e. all these processes use product structures somehow and there has to be consensus between the processes about the generic structures used. Configuration matrices can be seen as a way of communicating products, their structures and saleable options to the organization selling, building and maintaining the final products.

5.1.2. The role of the matrices in production

Production uses predefined modules and their structures to manufacture and assemble the product variants sold by the sales process. Companies use different kinds of software systems to control product structures, such as ERP and PDM, more effectively. During the sales delivery process the customer specific product individual is defined according to the customer specifications. This product structure is handled in the ERP system to provide the needed information to be used to build this product instance. The need for configuration knowledge maintenance grows even bigger if MRP is used as the main control philosophy of the production system. MRP requires that the product structures are flawless in order to decide right shop orders for manufacturing and right purchasing orders for purchasing. The inconsistencies found in ERP systems (see the definition in appendix 3) can be systematically detected when using generic product structures integrated with local ERP databases. Tools and analyses to be used in production are as follows:

- Invalid parts and components
- Parts and components that will become invalid as production of a certain product ends
- Parts and components that will become invalid as a feature of a certain product will become invalid
- Production cell analysis
- Analysis to define volume parts and components as well as modules and features
- Through put time analysis
- Platform analysis
- Inner platform analysis for product family
- Tools to help purchasing and manufacturing
    - active parts and components for a certain buyer and supplier
    - active parts and components for a certain manufacturing cell
    - active parts and components for a certain phase

All of the above analyses are based on the generic product structure of the selected product families to be analyzed. The main idea is that there are tools helping to keep all the stocks valid and to have effective means to figure out the current situation of the production system since as time goes by the risk of invalid parts and components in production obviously increases.

Next to the tools established the configuration matrices reveal the current state of modularity to provide insight into producibility in the context of modularity and configurability. This will be further discussed in 5.2. This will reveal the need for production based modularity and will integrate the product development and production development.

5.1.3. The role of the matrices in marketing

For marketing, configuration matrices are essential for determining the features offered to the customers. Features offered to customers are related to the options defined by the configuration matrices. These features and their options should be the same for production, marketing and also for product development.

The main advantage next to the marketing configuration rules is the possibility to use the configuration matrices for an algorithm in order to make feature based pricing possible. Configuration matrices show all the possible features and their options as well as all the modules according to the selected options and their possible combinations. The idea of the feature based pricing is to form a solid base by determining the level of material costs systematically using the valid configuration matrices and valid cost data from the ERP system. This enables the company to define right material costs for saleable options. The problem is that the options can form complex combinations between each other and without well-established configuration knowledge it is very troublesome to define right costs to be used as a basis for pricing. Considering the ERP databases there is a possibility to take material costs as well as labor costs to be included in pricing.

Next to the above-mentioned analysis and tools there is also a possibility to use ERP databases to define the usage of different features and their options. If the configuration matrices are integrated with the configurator, the saleable options are exactly the same in both places, i.e. it is possible to make matrix-based analyses to determine the frequency of the usage for certain options in a certain period of time. There is also another possibility as described in 3.8. When the configurator is established on top of the configuration matrices, the formed databases will provide many possibilities for analysis. These analyses include feature usage, price inflation and change frequency.

Finally, the sales-delivery process can be greatly improved by using a configurator. While valid configuration models provide only valid customer specific product individuals through the configurator to be produced, the time to configure orders can be shortened and the waiting times related to this process diminished. These error-free configurations will provide less deviation to the subsequent phases.

5.1.4. The role of the matrices in configurator maintenance

After the configuration knowledge has been documented it needs to be maintained. As time goes by and the life cycle of a product advances, changes to the product structure and to the configuration knowledge are likely to take place. Well-established change management processes are imperative to control and maintain the configuration knowledge, i.e. the changes from product design must be systematically handled in order to avoid inconsistencies in configuration models.

Configuration matrices offer a visual tool to handle the changing configuration knowledge. In the case of configurable products the configuration knowledge needs to be valid at all times regardless of the configuration process, i.e. the configuration knowledge needs to be valid both for manual and automatic configuration processes at all times. There are many possibilities for organizing the configurator maintenance, i.e. it can be the responsibility of production, product design or a configurator maintenance support organization. One of the main reasons to present configuration knowledge with matrices is the easy management of configuration rules in the changing environments. If the rate of change is frequent enough, the management of configuration rules in configurators can become impossible mainly because of the dependencies between saleable options. Tools and features that configuration matrices offer for configurator maintenance are:

- Interface to maintain the configuration matrices
    - generic structures
    - production configuration rules
    - marketing configuration rules
- Integration with configurator
- Visual representation of configuration knowledge
- Generic product structures
- Generic feature structures
- Base machines
- Feature families

Configuration matrices can be built manually by using Excel sheets or by using software system that can be developed for maintaining the matrices. The software system helps the user to build configuration matrices and confirms that the matrices are built systematically the same way in order for other software systems to interpret the matrices correctly. The same software system can be used to maintain the matrices as well. This software system has not been established yet.

Integration with the configurator is of great importance if the frequency of change is relatively high. In these cases the configuration knowledge is changing as product development makes changes to existing products. This now implies the need for versioning the matrices in order to secure the existence of configuration knowledge through the entire life cycle of a product, i.e. the configuration model needs to be sustained after the configuration task has been accomplished. For each configuration model and for each version of the model the configurator usually holds the data in the system. For long term management and in order to enable possibilities to reconfigure an existing product the need for maintaining the versions of configuration matrices is essential. While the product structures and features and their options change during the life cycle of the product, it would be very useful to integrate the configuration matrices with the configurator, i.e. all the features and elements needed to setup the configurator would automatically be loaded from the configuration matrices into the configurator. Configuration matrices hold all the knowledge both for the marketing configurator and also for the production configurator, i.e. both configuration models and configurator can be maintained in one place. This is also one of the reasons that the configurator maintenance team is developing into a direction of a cross functional team to serve many stakeholders in the company. The integration between configurator and configuration matrices is shown in Figure 57.

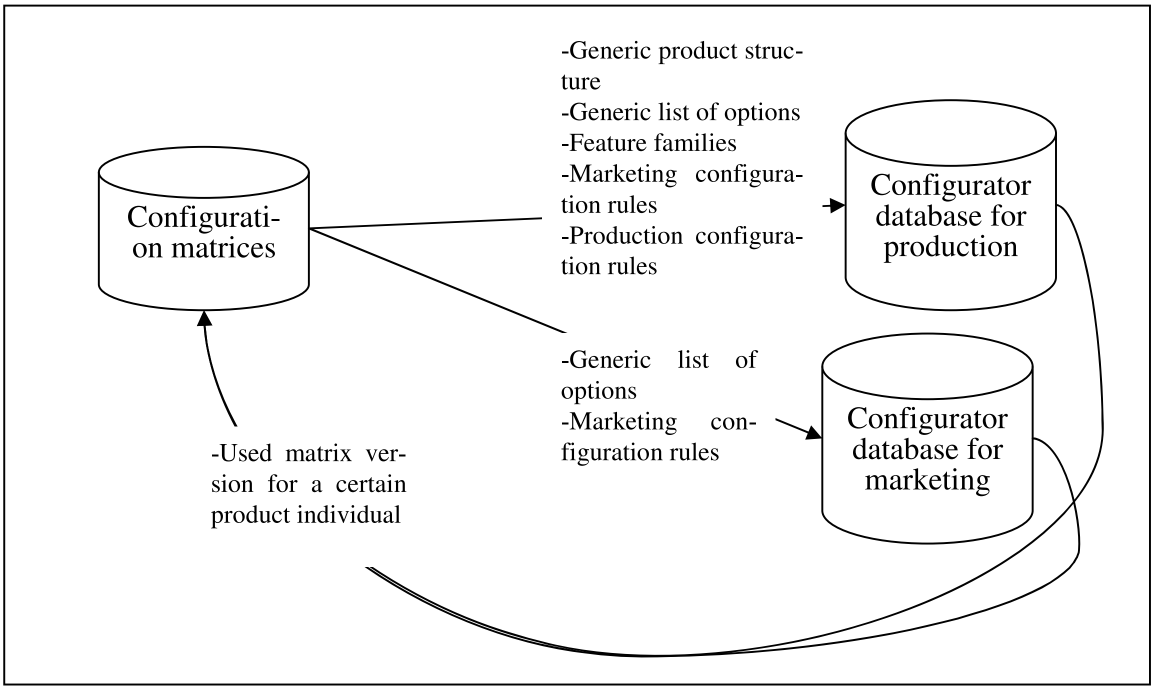


Figure 57. Integration between matrices and configurator

The above mentioned integration with the configurator provides effective means to update the configurator databases. One of the main obstacles is that conventional configurators are very inflexible when considering updating the databases automatically. This is the reason why section 3.8 considered another type of a configurator based directly on configuration matrices. The configurator presented in this thesis uses databases that are automatically updated and maintained. The configuration maintenance only updates the configuration matrices and the software system updates automatically the databases. Thus, the time consuming task of updating the configurator can be eliminated.

The rest of the features offered to configuration maintenance are mainly related to the product structure and its configurability. Even if product development should be responsible for creating configuration knowledge, the role of configuration maintenance increases when considering the evolution of modularity. The idea that configuration matrices can be used to understand modularity and configurability of a certain product and use the matrices to give an idea of the state of the modularity is discussed in the following section.

5.2. Producibility, modularity and configurability in the context of configuration matrices

The use of configuration matrices provided good understanding how producibility can be reached through modularity and configurability. The idea that modular product architectures should provide similarity between the product structure and the production system while simultaneously enabling configurability was clarified by using configuration matrices.

The product needs to support the production system, thus the platform part of the product architecture is important. Configuration matrices reveal the platform part of the product structure and provide insights into the possibilities to further develop the product structure. This is the reason why configuration matrices were tied to evolution model of modularity. Thus, when the company understands that there is a platform part for every configuration model and it can be decomposed freely (preferably according to production system) the cost related to configurable products can be lowered by concentrating on fast production system and adequate marshalling methods. Also delayed differentiation can be more easily concentrated on when the platform proportion of the product structure is well defined.

Next to the platform proportion the configuration matrices also include the variant modules and related features. This part is also important for producibility since it introduces deviation to the production system as the variant proportion or each sold customer specific product individual is different. The ability of product design to deliver producible products is related to its capability to provide adequate modularization in the case of configurable products.

All the above discussed benefits are related to the visualization and documentation of configuration knowledge. This is the main benefit related to configuration matrices provided by this thesis.

6. DISCUSSION AND FURTHER RESEARCH

This chapter discusses contribution provided by the research. Section 6.1 discusses the research questions and hypotheses presented in chapter 1, while section 6.2 discusses possibilities for further research.

6.1. Discussion of results

In this section configuration matrices and the results achieved are considered and the contribution discussed. The focus for showing the contribution for constructive research is to focus on discussing the usefulness and generalizability of the result (Olkkonen 1994). Also the novelty value is important. When considering validity and reliability of the result achieved during this thesis, the emphasis is to show how the core of the approach, i.e. configuration matrices provide repeatability through reliability as well as valid outputs for documenting product architectures and configuration knowledge. The reason for this is the research method used, i.e. constructive method has provided a solution to a problem and that solution method is considered next to the presented requirements to show robustness.

In chapter 1 the contribution was divided to provide more understanding both for the scientific community as well as for industrial applications next to the method and tools built during the research. This contribution is discussed in 6.1.1. In 6.1.2 the hypotheses presented in chapter 1 are discussed and in 6.1.3 the feedback from the case company, colleagues and conferences is considered. Finally generalizability, reliability and validity of the results are discussed in 6.1.4, 6.1.5 and 6.1.6 respectively.

6.1.1. Contribution for scientific community and industry

For the scientific purposes this study provides an insight to holistically consider modularity and configurability for the entire organization involved. The contribution for scientific community was summarized in chapter 1 as follows:

- Considering modularity and configurability in the context of producibility
- Providing a method to document configuration knowledge and use it to provide understanding of modularity and configurability
- Providing a framework for configuration knowledge based on established tools
- Providing requirements for a configurator in the changing environment

Modularity and configurability were considered deeply in the context of producibility. Configuration matrix representation with generic product structures was integrated with the modularity evolution model presented by Lehtonen (2003). This provided the connection between configuration models, modularity and its development. It was noted that the capabilities of product design affect the level of modularity provided and the types of modules it can produce. For producibility modularity can provide either benefits or lack of understanding providing unnecessary work to be done. Thus, modularity needs to be seen in the context of producibility.

The next issue that the configuration matrices were related to was the MBI method. Configurability is considered in MBI in a way that producibility and modularity are considered properly. The main idea that connects the MBI, configuration matrices and modularity is that configurability decides the largest possible size of the modules considering also producibility, thus modularization by integration. An easy way to document the configuration knowledge is provided for the MBI method by the configuration matrices. They also provide some requirements for modularity while simultaneously considering the configurator.

As mentioned during this work, producibility (in the case of configurable products) needs a configurable product which satisfies the needs of production systems, thus the product needs to be suitable for configuration and production simultaneously. Again, the similarity between the production system and product architecture is best accomplished with platform proportion, thus the MBI, modularization evolution and configuration matrices are integrated. Finally, configuration matrices were included into the MBI method which itself is a scientifically proved, accepted and working method. This provided means to build configuration matrices during the product design process.

The framework based on configuration knowledge and the tools established during this research provide further understanding how organizations can use configuration knowledge and the idea of generic product architectures.

Finally, for scientific contribution the requirements for a new configurator were established considering changing environments. First of all, the changes related to configuration models and tasks were defined and the requirements provided to support the processes related. The main issue was that the applicability of these requirements was established in a working configurator used in a real world application.

For industry the contribution was also considered in chapter 1. The suggested contributions were as follows:

- A clear and an easy method for documenting configuration knowledge
- Clear understanding of producibility, configurability, modularity and how these issues interact
- Clear understanding of the importance of configuration knowledge to organization
- Understanding the need for concurrent engineering to provide producible products
- Provide a configurator with minimal updating tasks
- Tools based on configuration knowledge

As mentioned before, the researcher was in a very close cooperation with the case company. For this reason the processes were very closely studied and the understanding was used to further develop the approach. For the contribution the main issue is that there is a very easy method that can be used to document the configuration knowledge. Next to the documentation there is also a very easy way of studying and maintaining the configuration knowledge. As mentioned during this work, the documentation of configuration knowledge is usually considered one of the main obstacles when implementing a configurator.

For more strategic issues this work also provides a very close consideration of producibility while modularity and configurability are very closely studied. The main focus is to provide understanding of these concepts so that the product architecture can be seen as an enabler through producibility to provide an effective production system. On the other hand, effective product architecture can be designed to support the existing production system. Producibility clarifies these thoughts. The need for concurrent engineering is also realized more easily when considering producibility. For a producible product modularity needs to be considered by taking all the important stakeholders into account.

The configurator established was considered to provide enough flexibility to be used in a changing environment. The main focus was to automate the most critical parts of configurator maintenance, thus the configuration rules are generated automatically and the validity of configuration revisions are also automatically considered. The main benefit for the industrial cases is the revisioning of configuration knowledge which mainly provides the flexibility needed. Also the tedious task of keeping the configurations updated with the latest module revisions was automatically handled.

Finally, the industrial cases can benefit from the tools based on configuration knowledge. Many of the tools are related to the configurator itself, but there are also tools that can be used based on the idea of generic product structures.

6.1.2. Verification of hypotheses

The hypotheses presented in chapter 1 were introduced considering the importance of documented configuration knowledge to the organization. There were three hypotheses presented considering configuration knowledge documentation and maintenance, possible automatic properties related and the importance of generic product structure. The hypotheses are represented below:

- Using configuration matrices, it is possible to present the generic product structures and the dependencies of the modules and saleable features so that the configurator itself and also the product structure can be maintained during the lifecycle of the product

- It is possible to develop a software system to interpret the matrices in order to automatically handle the updating task related to the configuration knowledge and to provide an effective product configurator

- There is a possibility to use the generic product structures and rules of the matrices to derive tools for analyzing the products and to use the configuration matrices as an integrative element for the organization

As there is no quantitative data available for proving or falsifying the hypotheses, approach presented by Olesen (1992) will be used. Olesen (1992) provides the following criteria to which the presented hypotheses can be considered to:

- **Internal logic** ("that the result is based on known and accepted theories, and that there is a connection between the starting point, hypothesis, and the result.")

The starting point is to document the configuration knowledge and to consider the importance and usability of this knowledge company wide in the context of producibility. Configuration matrices and related tools established during this research as a result can be used to address these issues. Thus, producibility in the context of modularity and configurability is considered. The hypotheses are supported by the results achieved through the developed method and the case study. Also the idea of generic product structures provided a useful tool to provide integration between different stakeholders. Thus, the case study provides a connection between the starting point, hypotheses and the results.

This work is resting on the case study and the state of the art provides the basis for the thesis. The state of the art provides methods and approaches, not solid theories. Thus, the case study is used to provide the verification as constructive research method requires. The internal logic in this sense is delivered by using the case study and the methods provided by the state of the art to deliver results based on hypothesis.

- **Truth** ("that the theoretical and practical result can be used to explain the 'real' phenomena")

The research process has been very iterative between the researcher and the case company. Currently the case company uses all the presented tools, also the configurator established is implemented. Documenting configuration knowledge with configuration matrices provided means for the case company to focus on the important issues related to producibility, modularity and configurability. While using the method provided by this research the case company could re-establish their ideas about product platforms, modularity and product configuration. The results and ideas were proved useful while widely used by the case company, thus they can be regarded to be true.

- **Acceptance** ("that other researchers accept the theories used in the project, and that professionals use tools based on the theory.")

Configuration matrices are widely used in the case company. Configuration matrices have reached a state where they are a part of the routine tasks when executing operative processes. For the scientific community configuration matrices have been presented in a conference (NordDesign 2004). Issues related to configuration matrices have been discussed with other researchers and they have accepted the ideas and considered using the approach for their projects when applicable.

- **Applicability** ("that the use of the tools allows the probability for success to increase with repeated use. It does not necessarily lead to success every time, but over a period of time will give better result than if not used.")

When considering configuration knowledge and its documentation, the exactness of this data is crucial for companies. Configuration matrices removed the need for tacit knowledge related to configurable products. Using configuration matrices provides a higher probability of success considering the configuration task. Other tools presented use the generic product structure to refine data for different applications. A solid basis for better results is then provided. A longer term issue is to handle producibility, modularity and configurability considering product design efforts. For this, the use of configuration matrices will provide guidance while the biggest benefits are only realized when using MBI, for example, with configuration matrices helping to document the configuration knowledge. MBI has proved to be applicable as well as configuration matrices have, thus applicability has been proven.

- **Novelty value** ("that new solutions are presented, or that new ways of looking at a problem are introduced.")

Considering configuration knowledge and generic product structures in the context of the entire organization to provide configuration knowledge based tools is a new way. Also the established configurator provides new features, not currently included in the conventional configurators. These features are automatic rules generator, automatic basis for pricing, change process algorithms to define validity of configuration knowledge, automatic reconfiguration of order books and automatic re-establishment of configuration knowledge after changes has been made to configuration models. Modularity and configurability were related and well discussed in the context of producibility and they were all related to configuration matrices. The criterion of novelty value has been reached.

6.1.3. Evidence of acceptance by feedback

The feedback considered in this section can be divided into feedback gathered from the case company, colleagues and conferences. The case company was central for verifying the use of configurator matrices as well as all the tools and ideas based on this approach. The main issues proving the usability of configuration matrices and presented ideas of producibility are summarized below:

- Personnel maintaining configuration knowledge and module interfaces concluded that this work would be close to impossible in the case company without the configuration matrices
- Marketing and sales use product configurator and other tools (automatic pricing, option frequency) based on configuration matrices
- Production personnel use production configurator to establish customer specific product structures
- Production development uses the generic product architectures based on configuration matrices frequently in their efforts
- Product development considers configurability and modularity more closely and more holistically
- Many additional projects were launched due to the deeper understanding of producibility related to configurability and modularity
    - considering new marshalling methods
    - considering platforms and product portfolio more closely
    - developing a new layout to better support the ideas of producibility

For the case company the benefits were distributed both for strategic level as well as to the operational level to aid both decision making as well as daily routines.

The feedback from colleagues was closely related to configuration matrices. The K- and V-matrix method was very closely benchmarked with the configuration matrices established during this research. Dr. Bongulielmi commented on the differences and also the similarities between the two matrix representations. Configuration matrices and used technologies are very close to the ones used in K- and V-matrices while the contribution lies in the framework as well as in the automatic features provided by the configuration rules generator. Thus, the novelty is realized from the wide use of configuration knowledge based tools as well as the automatic basis for pricing, rules revisioning and validity algorithms. Also colleagues in Tampere University of Technology researching modularity commented positively on the results of this study. They also noted that while Bongulielmi provided the first steps in configuration matrices, this type of method is also developed in other locations while the focus is slightly different.

Producibility, modularity and configurability were also the interest of different companies as well as different researchers during the Process-MSDD project. The topic of producibility was discussed frequently while this research provided good insights into configurable products. Especially modularity was considered by the research group in the context of producibility. Even if this research provided understanding how different types of modularity affect the production system, the conclusion by the group was that structural modularity should be the goal to provide producible products. Thus, the capabilities of product design including resources affect the type of modularity achieved which then again affects producibility. This is included into the idea that product family structure should be configurable while it simultaneously provides the similarity between production system and product structure. This can be visualized by the configuration matrices to provide understanding. Even though producibility is best reached with structural modules, companies in industry might be experiencing problems with different types of modularity than structural. Considering this research there is a possibility to understand producibility in the context of modularity and configurability to fix these problems. Thus, the group of researchers from Process-MSDD has approved the approach presented during this research.

Finally, one paper has been presented in NordDesign 2004. This paper (Nummela 2004) provided an insight into the configuration matrices considering also the impact of the configuration knowledge to the organization. Questions related to the topic were ranging from the realization of matrices for the case company to the benefits that customers can harvest from the systematic processes related to configuration matrices. Also the integration between MBI and configuration matrices was discussed. The feedback from the scientific community was encouraging since they approved the efforts made with such a close cooperation with industry.

6.1.4. Generalizability of results

The core of the entire approach is the use of configuration matrices to enable the use of the established tools and the configurator while simultaneously understanding producibility in the context of modularity and configurability. When configurable products are used, the documentation of the configuration knowledge is essential. Considering configurability the aim is to provide configurable products by using modularity. The modules are selected according to the features chosen, thus the configuration matrices can be used. If modularity is not utilized, parts and components can also be used as basis for configurable product in the case of simple product architectures. For both of the situations configuration matrices can be used. Thus, if there is a need to describe dependencies between two objects or combinations between many objects, configuration matrices can be used and the tools will follow. During the case study there were also configuration matrices established for a subsidiary company that produced computers and software systems for the final product. Matrices were easily established.

When considering different types of environments than make-to-order, the configuration processes can alter greatly. If there are needs to do design work for the sold customer specific product individual, the configuration matrices can still be used at least for the platform proportion as well as for the standard features with established modules. Thus, partial configuration is possible. The proportion that needs to be designed during the order-delivery process should be added to the configurator by adding matrix external features by enlarging the configurator software. When this type of configuration process is selected the follow-up processes need to be defined to the configurator and to the organization executing this process. This would be possible when establishing partial configuration and building processes to the product design to add modules, parts and components to the preconfigured product structure. The tools presented in the framework can be used whenever the generic product structure can be established.

Above, the configuration matrices and the established configurator have been considered. The case study considered producibility, modularity and configurability in the case of complex mechatronic products. The configuration matrices were used to understand modularity and configurability for the case company to provide deeper understanding how these issues are related to producibility. Producibility can be considered for any product and type of modularity. Even if products and their modularity and configurability for different companies can be dissimilar the main idea is that the features selected by the customer needs to be connected to the module level one way or another. While the features are connected to the module level the configuration matrices can be used to reveal the generic product structures. Concepts of producibility, modularity and configurability can be considered general in the context of this research while the use of configuration matrices will deliver the results presented in the case study of this research. The ideas of modular systems (Figure 15) and platforms next to producibility can be considered for any product. Thus, the presented method is not case specific and can be generalized to other applications and companies also.

Finally one of the main restrictions of the presented method is the software systems used by different companies. If there are no PDM or ERP systems available, the use of configuration matrices could be possible while the tools cannot be used. For these situations the use of a configurator is also questionable since the product structures are most likely ill-defined. In these situations configuration matrices can be used to document the current level of product structures and to lay the grounds for further development. One of the main points when using configurable product families is the documentation of configuration knowledge. When revealing and visualizing the configuration knowledge the configuration matrices can be used. If the product architecture cannot be described by configuration matrices there should be considerations how the product architectures should be decomposed to provide the benefits related to configurable products.

6.1.5. Reliability of results

Olkkonen (1994) sees that repeatability is the ability to repeat the research and to establish the same results with the same method. Reliability is a measure used for estimating the truthfulness of the gathered data (Olkkonen 1994). Olkkonen (1994) sees that reliability also considers the repeatability of the research. Thus, if the data gathered is reliable the results are repeatable. For the result of this research, reliability should be considered in relation to the provided approach to solve the problem of documenting configuration knowledge. Thus, the question is that when providing the same product architecture to different researcher, will they come up with a same type of documentation using the configuration matrices. This is the only measure for the approach since all the tools based on configuration matrices will follow automatically when the matrices are established appropriately.

The cases where configuration matrices are used to aid product design will include uncertainty due to the innovative nature of the process. For example, the results of the cabin case would be even more impressive if the entire architecture would have been re-engineered. Also the use of configuration matrices for these purposes needs a creative way of working and understanding the requirements for configurability and modularity simultaneously to provide wanted results. For the rest of the tools the requirement of repeatability is easily met by the use of the presented configuration matrices. The researcher will come up with the same results when using the configuration matrices since all the major tools have been established, thus only the configuration matrices need to be created. Naturally the results are repeated frequently by using these tools in the case company.

The lack of reliable data (problem with configuration matrices) is revealed by the frequent changes that configuration matrices experience after their establishment. This part is crucial for the approach since the tools are built entirely to the data provided by the configuration matrices. Thus, when configuration matrices have unreliable data also the results will be unreliable.

6.1.6. Validity of results

Olkkonen (1994) considers validity of results to be the consistency between the ability of the used gauge to measure what is supposed to be measured.

When considering the configuration matrices, the idea is that they provide understanding of product structure considering configurability and modularity. For the entire approach, i.e. the framework of configuration knowledge and the configurator, there is no other possibility for the configuration models than to be valid. Thus, if not valid, false customer specific product individuals will be produced. As mentioned, above frequent changes to configuration matrices provide an idea of the validity of the work done when establishing the configuration matrices. If the building phase of the configuration matrices is not done precisely and according to the processes presented in this work, the tools created will not provide the wanted results.

The next issue when considering the tools presented in this thesis, is the validity of data in ERP or PDM databases. Even if generic product structures are provided by configuration matrices and they are valid, the BOMs in the ERP or PDM databases need to be valid as well. This could be considered as one of the preconditions when even considering product configuration or modularity. Thus, configuration matrices are the gauge for configurability linking together producibility and modularity.

6.2. Limitations

The limitations of the presented approach are related to the limitations of the configuration matrices as well as to the limitations of the established tools. The limitations of the configuration matrices are:

- No numerical data or functions can be part of the plain matrices
- Describing module selections with multiple combinations is difficult
- No validity checks when updating configuration matrices
- No combinations between features possible

The main focus for the use of configuration matrices is to automate the processes related to updating the configuration knowledge for the configurator. For this reason all the needed information should be located into the configuration matrices to avoid any external data feeding to the configurator. Configuration matrices work only with AND and OR operators, i.e. "x" and features are used to determine the proper operator. Thus, no numerical data or functions are included into the matrices. Cases that need a selection of many modules of the same type due to the selection of other related modules is impossible to provide without the use of numerical data and functions. These types of tools should be included to the configurator software.

Describing the configuration knowledge in general is tedious. Even more so when the module selection includes many features and the combinations are complex. For this task a user interface should be established to provide an easier way of modeling configuration knowledge. Also when updating as well as creating configuration matrices manually the amount of data can mask errors. There are no validity checks to interpret the validity of configuration matrices. Thus, if these errors are present in the configuration matrices the tools established will also provide false outputs.

Finally, the configuration matrices do not provide any means to include dependencies between features. Thus, functions or similar configuration rules as modules have are needed for features. This is a problem of the configuration rules generator.

The limitations for the developed framework are as follows:

- After sales is not considered in depth
- The configurator does not support the use of multiple configuration models for one order

After sales was left outside of the scope of this thesis. The use of multiple configuration models for one order limits the use of the configurator. When this feature is established, the configurator will be able to handle inheritance of features as well. Currently the configurator is used to configure these different products separately while the order number ties all the products to one order. In order to provide this configurator to the sales force, it should include the use of multiple configuration matrices. For simple products the established configurator could be used without any modifications.

Considering producibility the limitations are related to company specific issues. When modularity and configurability are heavily studied, the implications can be seen all over the company. Changing the product architecture or production layout to support the ideas of producible products affects processes as well as the organization and the change can be either very slow or even impossible.

6.3. Further research

In this section the main directions for the further research related to the configuration matrices presented in this thesis are considered. The entire concept is based on configuration matrices, thus all the tools including the configurator are integrated with the matrices. There are three directions for further research:

- Configuration matrices and their representation
- Configurator
- The framework and its tools

The first direction is the development of a database that would ultimately replace the use of Excel as the main storage location for the matrices. This would need a well-defined user interface to maintain and update the configuration matrices. For this application the entire concept of configuration matrices would stay the same, only the database would be different. This would also require reprogramming of the configurator rules generator.

The second improvement emphasis should be targeted on the configurator itself. Following issues should be considered:

- Highest level matrix
- Inheritance of features

- Integration with CRM (customer requirements management)
- The possibility to provide a database for order quotations
- The possibility to provide functions
- Through put time analysis integrated to the configurator
- Capacity estimations for manufacturing

So far the configurator establishes the configuration task for one configuration matrix only. This can be fairly easily changed by using a higher level configuration matrix to provide the marketing configuration rules for different modular systems sold together. Thus, a configurator would use the highest level matrix first to determine the valid mix of products sold in one order. After this the features common to all the matrices would be presented, and after that all the configuration matrices would be treated individually by the configurator. There is already a software system ready to be used to provide the inheritance of features needed to provide the highest level configurator. Also the integration with a local CRM system needs to be accomplished next to the order quotation databases. Both of these tools are meant to be used to aid sales and marketing.

To provide functions to the configurator would enlarge its scope of usability. As the purpose of configuration matrices is to provide the configuration knowledge needed for the configurator automatically, the functions need to respect this assumption. The limitations of the matrix representation would be diminished by the use of functions to provide numerical data or, for example, functions between features. As of now, the features themselves are not allowed to make combinations in a way that modules and features do. Another possibility is to provide similar rules as modules have for the features. Thus, also the next features would be configured with a special configurator that uses the configuration rules for marketing in the same way that modules are determined currently. Finally, there is a possibility to include through put time analysis to the configurator. This would enable a real consideration of the warehouse situation to determine the earliest delivery date. Also the capacity estimations based on a configured product could be integrated with the configurator. For this application routing for manufacturing would be used and standard times would be established for the phases. Using these standard times the configurator would be able to provide capacity needs based on the configuration of a single product. This would need a tool suitable for determining the standard times as well as providing a database for storing the information.

Finally, the framework presented in Figures 20 and 21 should be further developed especially considering after sales. Figures 58 and 59 present the improved framework for configuration knowledge and the related tools. The diagonal of figures 58 and 59 includes all the tools presented in Figures 20 and 21 with the additional tools. The yellow areas in Figures 58 and 59 are new to the framework whereas the light blue areas include the additional tools (*in italic*) compared to Figures 20 and 21. These additional tools are concentrating on sales and marketing as well as production. Figures 58 and 59 show that heavy concentration is focused on the after sales to provide tools such as reconfiguration of old products. This could also include updating tasks to the newest revision of the configuration knowledge. Figures 58 and 59 also show the above mentioned issues related to the configurator itself. When the company uses multiple configuration models in one order, the inheritance and the option to build the highest level configuration matrix is required. Thus, only this will provide the configurator to be used for the sales force with the customer. The tools related to after sales include all the departments presented in Figures 58 and 59 while also purchasing and customer are considered.

STAKEHOLDER	CUSTOMER	MARKETING/SALES	PRODUCT DEVELOPMENT
CUSTOMER	Tools: *-Configurator* -Reconfiguration possibilities to update the machine based on new configuration models	Tools: *-Configurator*	Tools: -N/A
MARKETING/SALES	Tools: *-Configurator for sales* -Validity checks -Change processes -Price inflation -Option frequency *-CRM-link* *-Order quotation database*	Tools: -Marketing configurator -Option frequency analysis -Pricing software -Change processes -Validity checks -Price inflation analysis *-Configurator for sales* *-CRM-link* *-Order quotation database* *-Through put time analysis* *-Capacity estimations based on configured products* *-Life cycle cost for an option* *-Maintenance interval for options*	Tools: -Pricing software
PRODUCT DEVELOPMENT	Tools: -Analysis for marketing	Tools: -Analysis for marketing -Platform analysis	Tools: -Plafform analysis -Commonality of components, modules -Base machine analysis -Modularity development based on matrices -Change processes *-Maintenance interval for options* *-Module interface improvement for maintainability*
PURCHASING	Tools: *-New forecast system*	Tools: *-New forecast system*	Tools: *-New forecast system*
PRODUCTION	Tools: -Configurator *-Production leveling based on configured products*	Tools: -Configurator *-Production leveling based on configured products*	Tools: -Commonality of components, modules -Base machine analysis -Cell re-engineering
AFTER SALES	Tools: *-Reconfiguration*	Tools: *-Reconfiguration*	Tools: *-Matrices to support new product integration*

Figure 58. Further research related to tools for the framework

STAKEHOLDER	PURCHASING	PRODUCTION	AFTER SALES
CUSTOMER	Tools: -N/A	Tools: -N/A	Tools: *-Reconfiguration possibilities to update the machine based on new configuration models*
MARKETING/SALES	Tools: -Pricing software *-Through put time analysis*	Tools: -Pricing software *-Through put time analysis* *-Capacity estimations based on configured products*	Tools: *-Life cycle cost for an option* *-Maintenance interval for options*
PRODUCT DEVELOPMENT	Tools: -Commonality of components, modules -Base machine analysis -Platform analysis	Tools: -Commonality of components, modules -Base machine analysis -Plafform analysis	Tools: *-Maintenance interval for options* *-Module interface improvement for maintainability*
PURCHASING	Tools: -Active module based analysis -Through put time analysis -Invalid parts analysis *-New forecast system* *-Spare part order optimization*	Tools: -Active module based analysis -Delivery time window -Through put time analysis	Tools: *-Spare part order optimization*
PRODUCTION	Tools: -Delivery time window -Active module based analysis	Tools: -Through put time analysis -Invalid parts analysis -Cell re-engineering -Plafform analysis -Product kill analysis -Base machine analysis -Change processes -Production configurator -Validity checks *-Production leveling based on configured products* *-Capacity estimations for spare parts based on generic product structure*	Tools: *-Capacity estimations for spare parts based on generic product structure*
AFTER SALES	Tools: *-Product kill analysis*	Tools: *-Product kill analysis*	Tools: *-Reconfiguration* *-Matrices to support new product integration* *-Product kill analysis*

Figure 59. Further research related to tools for the framework continued

7. SUMMARY

This research has been concentrating on configuration knowledge, its representation, modeling and impact to the organization. Documentation and visualization of the configuration knowledge is of great importance for companies designing and manufacturing configurable products. Configurable products have an impact throughout the organization while the amount of knowledge related to these product families serve as a basis for effective processes. Configuration knowledge integrates the organization.

Modularity is often considered as a basis for configurable products while producibility includes the aspect of modularity and configurability. Modularity provides a basis for configurability while also other requirements need to be fulfilled. Production needs the similarity between the product structure and the production system in order to be able to operate efficiently. For the product family to be producible in the context of this research it needs to provide configurability and similarity between the production system and the product structure simultaneously. In order to provide these features the product family needs to be modular while the modularity needs to be designed into the product family at early stages of the product development process.

The emphasis of this research has been to establish a solid way of representing the configuration knowledge and to provide a framework that establishes different types of tools that can be used in several applications throughout the organization. Configuration matrices were established to provide an easy way of documenting the configuration knowledge related to the configurable products. These matrices provided a way to visualize the rules and dependencies of the product family structure. The configuration matrices were connected to the evolution of modularity and to the product development processes to provide producible product families.

The framework includes tools based on the generic product structure defined by the configuration matrices that provide information which can be otherwise hard to gather. The framework also includes a configurator that provides means to handle the configuration task in the changing environment. The idea is that the knowledge presented in the configuration matrices is generated automatically to be used as a basis for the configurator. Thus, the manual work of maintaining the configurator has been shifted to maintaining the configuration matrices.

While configurable products are used to satisfy the customer needs, changes occur during the lifecycle of the product family as well as during the lifecycle of a customer order. Both situations are related to the usability of the configurator since whenever the configuration knowledge is changed also the configurator is affected. Thus, there needs to be a way to decide the state of the configuration knowledge used in the configurator. For this reason three checking algorithms were established. These algorithms were based on the revisioned configuration knowledge based on the configuration matrices that evolve as time passes. The main benefits for this type of a configurator are the fact that changes to already configured products can be established during the order-delivery-process and the changes to the configurable products from product development can be effectively implemented. Thus, the changes to the configuration matrices can be evaluated with the algorithms and the order books automatically reconfigured to include the latest product structures when appropriate. If the configuration knowledge, i.e. the configuration matrices have been changed so that the algorithms fail, the configured product individuals need to be reconfigured. The configurator established is concentrating on minimizing the work related to the configurator maintenance as well as to provide means to include flawless information to the organization.

Finally, the case study was used to show the usability of the presented approach. The tools presented in the framework are all used in the case company. The most critical part of the framework for the case company was the configurator while the tools based on generic product structure were used to provide accurate information for many stakeholders. The configurator and the tools established are mainly operational issues while more strategic ones are producibility, modularity and configurability. These three factors integrated with the configuration matrices helped the case company to understand and to take actions related to the product family architectures. As a matter of fact, they have established a new basis for a platform with a step-by-step approach provided by the configuration matrices and the revealed configuration knowledge. The maintenance of configuration knowledge and the aspect of producibility have become well known for the case company's employees.

8. REFERENCES

Aarnio J., 2003. Modularization by Integration: Creating Modular Concepts for Mechatronic Products. Doctoral thesis, Tampere University of Technology, Publications 421, Tampere. ISBN 952-15-1027-7. [157 p.]

Aasland K., Reitan Jarl., Blankenburg Detlef., 2001. Even Modular Products can be Unmanageable. Proceedings of the 5th WDK Workshop on Product Structuring, Tampere. [8 p.]

Aldanondo M., Rouge S., Veron M., 2000. Expert configurator for concurrent engineering: Cameleon software and model. Journal of Intelligent Manufacturing 11 (2000). pp. 127-134.

Baldwin C., Clark K., 1997. Managing in an Age of Modularity. Harvard Business Review, Sep/Oct97, Vol. 75 Issue 5. pp. 84-94.

Browning T., 1998. Use of Dependency Structure Matrices for Product Development Cycle Time Reduction. Proceedings of the fifth ISPE International Conference on Concurrent Engineering: Research and Applications, Tokyo, July 15-17, 1998. [8 p.]

Bongulielmi, L., 2004. The K- & V-Matrix. Internet site referred [2.2.2004]. <URL: http://www.coma.imes.ethz.ch/>

Bongulielmi, L., 2002. Die Konfigurations & Verträglichkeitsmatrix als Beitrag zur Darstellung konfigurationsrelevanter Aspekte im Produktentstehungsprozess. Doctoral thesis Nr. 14904, Zurich University of Technology, Zurich. [218 p.]

Bongulielmi, L., Henseler, P. Puls, C., Meier, M., 2002. The K- & V-matrix method in comparison with matrix-based methods supporting modular product family architectures. Proceeding of NordDesign 2002, Trondheim. [8 p.]

Bongulielmi, L., Henseler, P. Puls, C., Meier, M., 2001. The K- & V-matrix method – An Approach in Analysis and Description of Variant Products. Proceeding of International Conference on Engineering Design ICED 01, Glasgow, August 21-23, 2001. [8 p.]

Dahmus J., Gonzalez-Zugasti J., Otto K., 2001. Modular Product Architecture. Design Studies Vol 22 No.5. pp. 409-424.

Eppinger S., Whitney D., Smith R., Gebala D., 1994. A Model-based Method for Organizing Tasks in Product Development. Research in Engineering Design Vol. 6 No. 1. pp. 1-13

Erixon G., 1998. Modular Function Deployment – A method for Product Modularization. Doctoral thesis, The Royal Institute of Technology, Stockholm. [178 p.]

Erixon G., Erlandsson A., Yxkull A., Östergren B., 1994. Modulindela Produkten – halverade ledtider och offensiv marknadsorientering. Sveriges Verkstadsindustrier, Stockholm. [216 p.]

Forza C., Salvador F., 2002a. Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise. Journal of Computers in Industry 46 (2002). pp. 37-46.

Forza, C., Salvador, F., 2002b. Managing for variety in the order fulfillment and acquisition process: The contribution of product configuration systems. International Journal of Production Economics 76 (1). pp. 87-98

Gershenson J., Prasad G., Zhang Y., 2004. Product modularity: measures and design methods. Journal of Engineering Design Vol. 15 No. 1 (2004). pp. 33-51.

Hansen P., Mikkola J., 2004. Platforms and architectures. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [9 p.]

Holmqvist T., Persson M., 2004. Modularization - not only a product issue. Proceedings of the 7th Workshop on Product Structuring - Product Platform Development, Gothenburg, March 24-25, 2004. [14 p.]

Hölttä K., 2004. Comparative analysis of product modularisation. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [9 p.]

Jensen T., Hildre H., 2004. Design for variety: a review in methods to establish a product family architecture. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [10 p.]

Jorgensen K., 2001. Product configuration- concepts and methodology. Proceedings of the fourth SMESME International conference, Aalborg, Denmark, 2001. pp. 314-322

Juuti, T., Lehtonen, T., Pulkkinen, A., Riitahuhta, A., 2004. Efficient Platform Utilisation with Configurable Products – Enabling factors illustrated. Proceedings of the 7th Workshop on Product Structuring- Product Platform Development, Chalmers University of Technology, Gothenburg, March 24-25, 2004. [12 p.]

Juuti, T., Lehtonen, T., 2004. The allocation of platform investments for more competitive and profitable products. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [10 p.]

Järvinen P., 2001. On Research Methods. Opinpajan kirja, Tampere. ISBN-951-97113-9-2. [190 p.]

Lapinleimu I., 2002. Tuotettavuuden olemus. Itä-Suomen ideaalitehdas-projektin teemapäivä, Kuopio, 5.11.2002. In Finnish. [12 p.]

Lapinleimu I., 2000. Ideaalitehdas – Tehtaan suunnittelun teorian kiteytys. Doctoral thesis. Tampere University of Technology, Institute of Production engineering. Institute report 50. In Finnish. [197 p.]

Lancaster K., 1990. The Economics of Product Variety: A Survey. Marketing Science, Vol. 9, No. 3 (1990). pp. 189-206

Lehtonen, T., Juuti, T., Pulkkinen, A., Riitahuhta, A., 2004. Dynamics of Product Platform Lifecycle and Disposition Management. Proceedings of the 7th Workshop on Product Structuring- Product Platform Development, Chalmers University of Technology, Gothenburg, March 24-25, 2004. [7 p.]

Lehtonen, T., Juuti, T., Pulkkinen, A., Riitahuhta, A., 2003. Dynamic Modularization – A Challenge for Design Process and Product Architecture. Proceedings of International Conference on Engineering Design ICED 03, August 19-21, Stockholm, 2003. [10 p.]

Malmqvist, J., 2002. A Classification of Matrix-based Methods for Product Modeling. Proceedings of International Design Conference – Design 2002, Dubrovnik, 2002. pp. 203-210.

Marshall R., Leaney P., 1999. A systems engineering approach to product modularity. Loughborough University, Loughborough. [6 p.]

Martin M., Ishii K., (2000). Design for Variety: a methodology for developing product platform architectures. 2000 ASME Engineering Technical Conferences, September 10-13, Baltimore, USA, ASME. [15 p.]

McCutcheon D., Raturi, A., Meredith R., 1994. The customization- Responsivenenss Squeeze. Sloan Management Review, Winter 1994. pp. 89-99.

Mesihovic S., Malmqvist J., 2004. A Process-Oriented Approach for Management of Product Configuration Models. Proceedings of the 2004 ASME international DETC/CIE: The Computers & Information in Engineering Conference, Salt Lake City. [22 p.]

Meyer M., Lehnerd A., 1997. The Power of Product Platforms – Building Value and Cost Leadership. The Free Press, New York. [267 p].

Mikkola J,. 2003. Modularization in New Product Development: Implications for Product Architectures, Supply Chain Management, and Industry Structure. Doctoral thesis, Copenhagen Business School, Ph.d. –Series 3.2003. [348 p.]

Mittal S., Frayman F., 1989. Towards a generic model of configuration tasks. International Joint Conference on Artificial Intelligence, pp. 1395-1401

Muffato, M., Roveda M., 2002. Product architecture and platforms: a conceptual framework. Int. Journal of Technology Management, Vol. 24, No. 1, pp. 1-16.

Muffato, M., Roveda M., 2000. Developing product platforms: analysis of the development process. Technovation 20 (2000). pp. 617-630.

Muffato, M., 1999. Introducing a platform strategy in product development. International Journal of Production Economics 60-61 (1999). pp. 145-153.

Muffato, M., 1997. Enhancing the product development process through a platform approach. In: Proceedings of 14th international conference on Production Research, Osaka, August 4-8, 1997. pp. 1224-1227.

Männistö T., Peltonen H., Alho K., Sulonen R., 1993. A Framework for Long Term Information Management of Product Configurations. Publications of Helsinki University of Technology, Department of Computer Science, Helsinki [13 p.].

Männistö T., Peltonen H., Sulonen R., 1996. View to Product Configuration Knowledge Modeling and Evolution. Publications of Helsinki University of Technology, IIA Research Centre and Laboratory of Information Processing Science, Helsinki. [8 p.]

Männistö T., Peltonen H., Sulonen R., 1995. Open Data Modeling Issues in Product Configuration. Publications of Helsinki University of Technology, Department of Computer Science, Helsinki, 1995. [11 p.]

Nummela, J., 2004. Matrix based method for maintaining configuration knowledge. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [10 p.]

Nummela J., 2003. Development of modularity of configurable products. Masters thesis, Tampere University of technology, Tampere. In Finnish. [96 p.]

Olesen J., 1992. Concurrent Development in Manufacturing – based on dispositional mechanisms. Ph.D. Thesis, Institute of Engineering Design, Technical University of Denmark. [154 p.]

Olkkonen T., 1994. Johdatus teollisuustalouden tutkimustyöhön. Helsinki University of Technology, Department of Industrial engineering, Helsinki. ISBN 951-22-1774-0. [143 p.]

Pahl G., Beitz W., 1986. Koneensuunnitteluoppi. Suomen Metalli-, Kone- ja sähköteollisuuden Keskusliitto, MET, Helsinki. In Finnish. [608 p.]

Parkinson B., Hudson P., Senior R., Short C., Barker R., 1996. A Virtual Environment for Concurrent Engineering using Artificial Intelligent Techniques. CEE'96 Conference, Robinson College Cambridge, 10-12 April 1996. [6 p.]

Partanen J., Haapasalo H., 2004. Fast production for order fulfillment: Implementing mass customization in electronics industry. International Journal of Production economics 90 (2004). pp. 213-222.

Peltonen H., Männistö T., Soininen T., Tiihonen J., Martio A., Sulonen R., 1998. Concepts for Modeling Configurable Products. Publications of Helsinki University of Technology, TAI Research Centre and Laboratory of Information Processing Science. [12 p.]

Pine, J, 1993. Mass Customization – The New Frontier in Business Competition. Harvard Business School Press, Boston. [333 p.]

Pulkkinen, A., Lehtonen, T., Riitahuhta, A., 2004. Characterizing Modular Product Families for Configuration. 7th Workshop on Product Structuring- Product Platform Development, Chalmers University of Technology, Gothenburg, March 24-25, 2004. [12 p.]

Pulkkinen A., Bongulielmi L., 2004. Design of product families for configuration. Proceedings of NordDesign 2004 – Product Design in Changing Environment 18-20 August 2004, Tampere, Finland. [10 p.]

Riitahuhta, A., 2000. Views and experiences of configuration management, in Riitahuhta A., Pulkkinen A., Design for configuration, a debate based on the 5th WDK workshop on product structuring, Tampere, 2000. Sringer Verlag, Berlin, 2001, ISBN 3-540-67739-9. [222 p.]

Robertson, D., Ulrich, K., 1998. Platform Product Development. Sloan Management Review, Summer 1998. [33 p.]

Salvador F., Forza C., 2004. Configuring products to address the customization- responsive squeeze: A survey of management issues and opportunities. International Journal of Production economics 91 (2004). pp. 273-291.

Salvador F., Forza C., Rungtusanathmam M., 2002. Modularity, product variety, production volume, and component sourcing – theorizing beyond generic prescriptions. Journal of Operations Management 20 (2002). 549-575.

Sekolec R., Kunz A., Meier M., 2003. Methodology for product structuring in the early stages of the design process. Proceedings of International Conference on Engineering Design ICED 03, Stockholm, August 19-21, 2003. [9 p.]

Selladurai R., 2004. Mass customization in operations management: oxymoron or reality? Omega 32 (2004). pp. 295-300.

Stake R., 1999. A hierarchical classification of the reasons for dividing products into module – A theoretical analysis of module drivers. Licentiate thesis. Royal Institute of Technology, Stockholm 1999. [65 p.]

Steger-Jensen K., Svensson C., 2004. Issues of mass customisation and supporting IT-solutions. Journal of Computers in Industry 54 (2004). pp. 83-103.

Steward D., 1981. The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transactions on Engineering Management, vol. 28, s. 71-74.

Stone R., Wood K., Crawford R., 1998. A heuristic method to identify modules from a functional description of a product. Proceedings of DETC98, 1998 ASME Design Engineering Technical Conferences, Atlanta, September 13-16, 1998. [11 p.]

Suri, R, 1998. Quick Response Manufacturing. Productivity inc., Portland. [544 p.]

Tiihonen, J., Soininen, T., Niemelä, I., Sulonen, R., 2003. A practical tool for mass-customising configurable products. Proceedings of International Conference on Engineering Design ICED 03, Stockholm, August 19-21, 2003. [10 p.]

Tiihonen J., 1999. Kansallinen Konfigurointikartoitus – asiakaskohtainen muuntelu suomalaisessa teollisuudessa. Licentiate thesis. Helsinki University of Technology, Department of computer sciences, Helsinki. In Finnish. [193 p.]

Tiihonen J., Lehtonen T., Soininen T., Pulkkinen A., Sulonen R., Riitahuhta A., 1998. Modeling Configurable Product Families. 4th WDK Workshop on Product Structuring, Delft, October 22-23, 1998. [22 p.]

Tiihonen J., Soininen T., 1997. Product Configurators – Information system support for configurable products. Technical report TKO-B137, Helsinki University of Technology, Laboratory of Information Processing Science. [22 p.]

Torvinen S., Britton G., Cochran D., 2004. Collective system design toolset – Instruction manual. Tampere University of Technology, Institute of Production Engineering. [152 p.]

Torvinen S., Britton G., Lapinleimu I., 2003. Systematic Design for Producibility. Tampere University of Technology, Institute of Production Engineering. Institute report 63. In Finnish. [54 p.]

Ulrich K., Eppinger S., 2000. Product Design and Development. The McGraw – Hill Companies, Inc., New York. [358 p.]

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. Research Policy 24 (1995). pp. 419-440.

Ulrich K., Tung K., 1991. Fundamentals of Product Modularity. Proceedings of the 1991 ASME Winter Annual Meeting Symposium on Issues in design/manufacturing integration, Atlanta USA, 1991. pp. 73-79.

Whitney D., 1992. Systematic design of modular products at Telemechanique. Internet site referred [6.7.2004] <URL: http://web.mit.edu/ctpid/www/Whitney/europe.html> [6 p.]

Österholm J., Tuokko R., 2001. Systemaattinen menetelmä Tuotemodulointiin – Modular Function Deployment. Metalliteollisuuden Keskusliitto, MET, Helsinki. In Finnish. [64 p.]

APPENDIX 1

Process-MSDD in short

The Process-MSDD project is a continuum of the long-term research conducted at Tampere University of Technology at the Institute of Production Engineering. The basis for the Process-MSDD project has been built in numerous earlier projects such as:

- Ideal Factory,
- Highly Productive and Reconfigurable Manufacturing Systems (HIPARMS),
- Modelling and Simulation of Manufacturing and Assembly times and Costs (MODMAC),
- Modelling of Dynamic Production Networks (MODNET),
- PlaNet One (Planning and control of Networked Production systems, 1), and
- Planet Two.

The Process-MSDD project is concentrating on further developing the MSDD (Manufacturing System Design and Decomposition) model based on the philosophy developed at Massachusetts Institute of Technology (MIT), USA. The MSDD model has been developed by Dr. David S. Cochran (Associate Professor at MIT before establishing his own consultancy company System Design LLC) and the staff of MIT's Production System Design Laboratory (PSD). The original MSDD model has been extended through the subsequent research conducted by Dr. Cochran and Professors Graeme A. Britton (Nanyang Technological University, Singapore) and Seppo Torvinen (Tampere University of Technology, Finland). The cooperation has produced a model based on the MSDD concepts called CSD (Collective System Design) that includes the original MSDD model, the so called 5S tree (Japanese principles on the visual workplace), and the Product design for producibility tree.

This thesis has been considering producibility in the context of modularity and configurability, thus providing a case study for the larger project.

The main goals of the Process-MSDD project are as follows:

- Including a process view and transparency to the MSDD models
- Inserting resolution tools and methods into the models
- Integrating change processes and their management into the models
- Updating the metric and key indicator system
- Conducting two large scale industrial cases
- Development and update of the CSD software portal on the basis of feedback from the industrial case studies

The main idea is that the researchers are located into the case companies to study their field of interest. Thus, the scientific community benefits from the industrial feedback and the usage of MSDD based models while the case companies benefit from using the tools as well as getting the latest research available through the researchers as well as through the

Professors. The case companies also get licences for the CSD-portal to be used during and after the research.

Publications related to the results of the MSDD-based models can be found from the following reports:

Torvinen S., Cochran, D., Lapinleimu I., 2002. An Architecture for Product Design for Producibility. Tampere University of Technology, Institute of Production Engineering. Institute report 58. ISBN 952-15-0824-8

Torvinen S., Cochran, D., Lapinleimu I., 2002. An Architecture for the Design of Manufacturing System Structure. Tampere University of Technology, Institute of Production Engineering. Institute report 59. ISBN 952-15-0825-6

Torvinen S., 2002. Tuotantojärjestelmien rakenteinen ja ositettu suunnittelumalli. Manufacturing System Design and Decomposition (MSDD). Tampere University of Technology, Institute of Production Engineering. Institute report 61. ISBN 952-15-0865-5. In Finnish.

Torvinen S., Britton G., Lapinleimu I., 2003. Systematic Design for Producibility. Tampere University of Technology, Institute of Production Engineering. Institute report 63. In Finnish.

The latest advancements made to the models are reported in:

Torvinen S., Britton G., Cochran D., 2004. Collective system design toolset – Instruction manual. Tampere University of Technology, Institute of Production Engineering. [152 p.]

APPENDIX 2

The following pseudo code is the representation of the first algorithm used to check the validity of configuration knowledge for change situations presented in Figure 34.

```
FUNCTION CHECK_OPTIONS ()
        DECLARE VARIABLES
        SET CONNECTIONS
        SET RECORDSETS
        DEFINE CONNECTION STRING
        OPEN CONNECTION
        'IF THE REVISION EQUALS REVISION MINUS 1 RETURN THAT OPTION STRUCTURE
        'HAS NOT BEEN CHANGED
        IF REVISION <> REVISION - 1 THEN
                DEFINE SQL-STRING WHERE THE REVISION IS REVISION - 1
                OPEN RECORDSET 1 WITH THE SQL-STRING
                DO WHILE END OF FILE IS FALSE
                              DEFINE SQL-STRING WHERE THE REVISION IS BETWEEN
                              REVISION AND REVISION - 1
                              OPEN RECORDSET 2 WITH THE SQL-STRING
                              FOUND = FALSE
                              DO WHILE END OF FILE IS FALSE
                                      IF OPTION 1 = OPTION 2 THEN
                                           FOUND = TRUE
                                           EXIT
                                      END IF
                                      MOVE TO THE NEXT RECORD
                              LOOP
                              CLOSE RECORDSET 2
                              IF FOUND = FALSE THEN
                                      'DID NOT FOUND MATCHING OPTION
                                      EXIT
                             END IF
                             MOVE TO THE NEXT RECORD
                LOOP
                CLOSE RECORDSET 1
                IF FOUND = FALSE THEN
                             'RETURN THAT OPTION STRUCTURE HAS BEEN CHANGED
                             CHECK_OPTIONS = FALSE
                ELSE
                             'RETURN THAT OPTION STRUCTURE HAS NOT BEEN CHANGED
                             CHECK_OPTIONS = TRUE
                END IF
        ELSE
                'RETURN THAT OPTION STRUCTURE HAS NOT BEEN CHANGED
                CHECK_OPTIONS = TRUE
        END IF
        SET RECORDSETS NOTHING
        SET CONNECTION NOTHING
END FUNCTION
```

APPENDIX 3

Tools based on configuration matrices

This appendix will present the established tools based on configuration matrices. The configurator, pricing, change processes and validity checks for configuration matrices have been deeply considered in section 3.8 and not discussed here. The tools presented below are consistent with the presented framework for configuration matrices and are used by the case company.

Price inflation

The pricing software system presented in section 3.8 forms the basis for estimating price inflation for different features. Price inflation means that the costs for manufacturing and producing the feature have been increased for some reason. When the configurator databases are updated with the configuration rules generator, also the costs for features are automatically re-established. The interface presented in Figure 33 illustrates the mean value with the maximum and minimum values for features. These values change if there has been a change in the configuration knowledge (e.g. module interfaces have changed) or parts and components prices have gone up. Obviously the selling price of a feature used by the configurator stays the same even if the configurator databases are updated.

The pricing database also stores invalid prices for features. From this knowledge the price development for features can be estimated.

Option frequency

For marketing and sales the frequency of options sold is valuable. This information is valuable also for product development when considering features to be included into new products. From Figures 48 and 49 the module frequencies were estimated using the configuration matrices and established configurations stored in ERP databases. Option frequency can also be estimated by using the information from Figures 48 and 49 by comparing the modules with the configuration matrices. Thus, the options of the modules can be determined and the frequency estimated.

Another way to do this is to use the configurator databases. While all the produced configurations are stored in the databases they can be easily compared and option frequency estimated. Also ERP databases can be used to estimate the option frequency if the options selected are stored in the system. For the case company this type of estimation was done using 94 latest configurations found in the system. The result for the bigger harvester machine Ergo is presented in Figure 60.

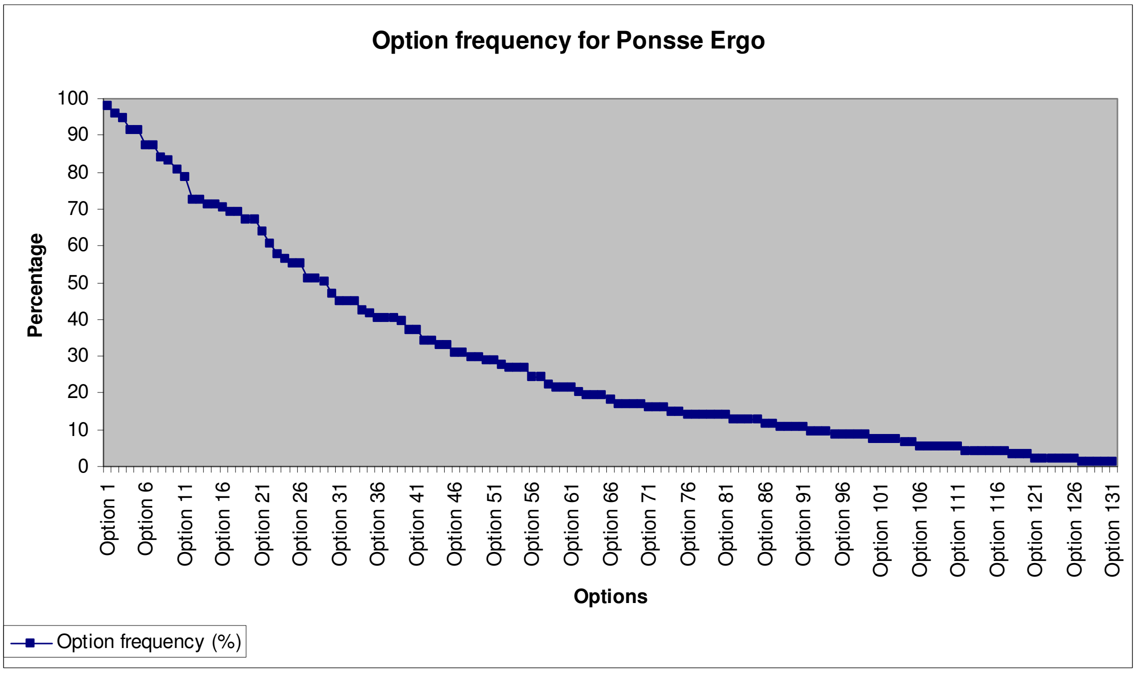


Figure 60. Option frequency for Ponsse Ergo

Figure 60 presents produced configurations and for these configurations 131 different options were selected to be included in one or more configurations. Similar types of charts can easily be derived from the configurator databases after enough products have been configured.

Through put time analysis

Through put time analysis can be performed to predict the critical path for a product or for a set of parts or components. Thus, the delivery time for a configured product can be estimated (Steger-Jensen and Svensson 2004). For a configurable product, the lead time for a configuration depends on the lead times of the parts, components and modules to be purchased, manufactured or assembled. Figure 61 presents the idea of through put time analysis (Torvinen 2003).

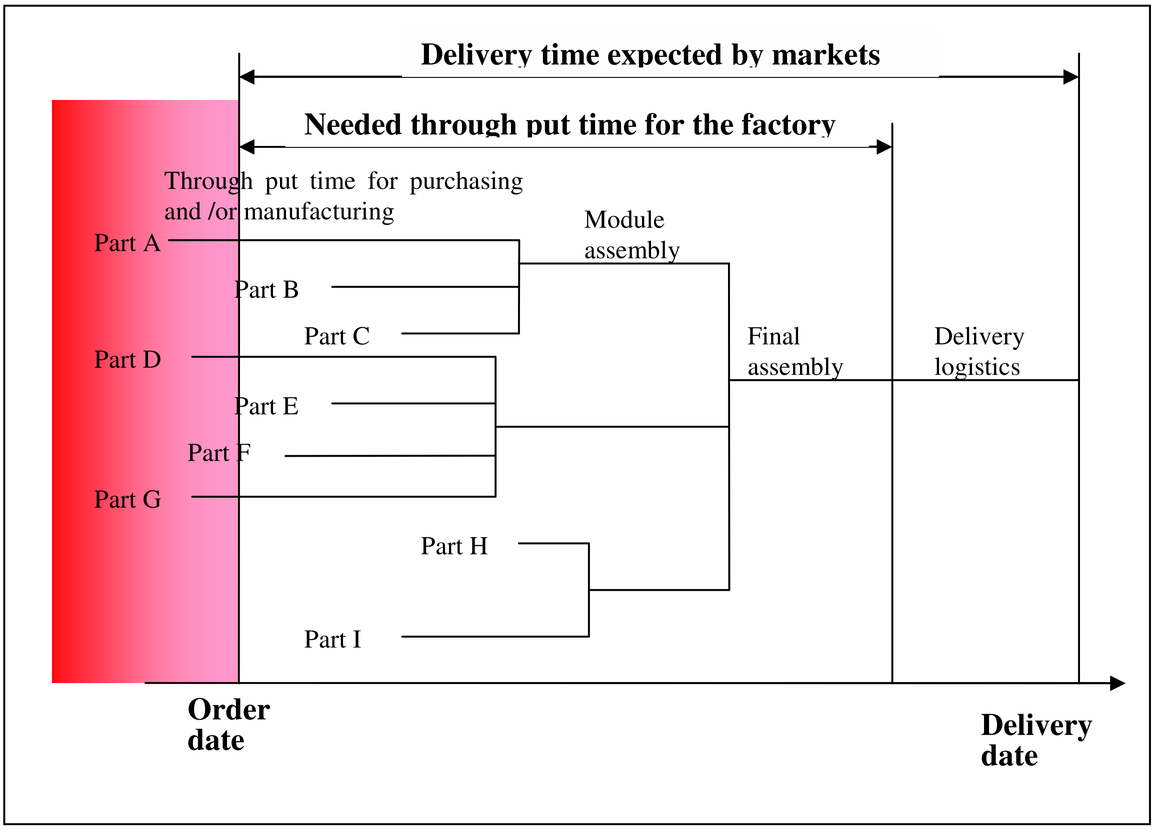


Figure 61. Basis for through put time analysis (Torvinen 2003)

The basic idea is that the lead times for production system and purchasing are analyzed and parts and components that do not fit the needed delivery time requirements need to be reconsidered. This is one requirement for a producible product. According to Torvinen (2003), the following questions can be asked:

- Can the product be changed so that materials, parts and modules which have long lead times will not be needed?
- Can the supplier be changed to a supplier with shorter lead times?
- Can the processes having long lead times to be improved so that they can meet the delivery time window requirements?
- Can the parts be buffered (sufficient yearly volume and repeatability) without the need for forecasts?

If the answer to all above question is no, the material, part or module needs to be stored in a warehouse. Also the marshalling of the material, part or module needs to be done by forecasts or by push-control (e.g. MRP) (Torvinen 2003).

The idea presented in Figure 61 can be considered to the entire production system. This now helps the purchasing and manufacturing departments to better locate their development efforts. Also the product development needs to be available if re-engineering of the product is necessary. For the case company through put time considerations and the importance of short lead times were understood. The result was that additional projects were launched to streamline the production and to consider the processes for purchasing very closely.

Considering the configuration matrices the through put time analysis uses the active parts and components from the generic product structures and automatically provides the user with the through put time analysis. This can be used by a manufacturing, purchasing or product development engineer when considering lead times for a part, component or module.

For the configurator, through put time analysis provides an interesting tool to be added. As pricing is an automatic part of the configurator, so could the lead time estimations for a specific configuration be possible as well. This would require the configurator to check all the available parts and components with reservations to other products and to derive a first possible delivery date for the configured product. For the case company this has not been established yet because processes related to purchasing, forecasting and manufacturing are not in sufficient levels so that providing this type of information would be of any help.

Platform analyses

As presented in section 4.2.2, the platforms for configuration matrices for existing products were verified by platform analysis. Two types of analyses were used: based on configuration matrices and existing configurations of produced products. Platform analysis for configuration matrices can be used to analyze platforms inside a modular system or between modular systems, i.e. between different configuration matrices. An interface for platform analyses is presented in Figure 62.

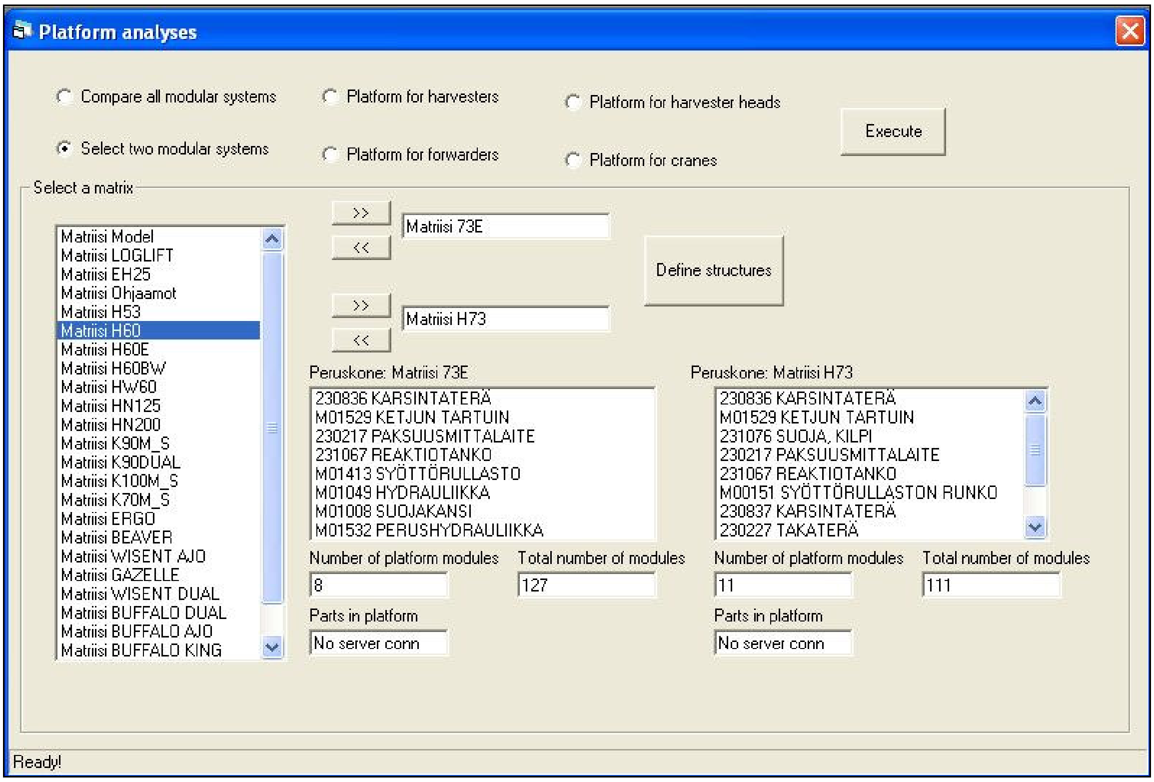


Figure 62. The interface for platform analysis

These analyses include only the standard part of the modular system and the comparison is made between two or more systems to figure out the platform for standard proportions of a product. In Figure 62 two modular systems are compared. For the case company there are also possibilities to define platforms for (Figure 62):

- All the modular systems
- Harvester machines
- Forwarder machines
- Harvester heads
- Cranes

For the cabin there is only one configuration matrix, i.e. no comparison with other systems is meaningful. When selecting two modular systems for comparison, the cabin can be considered next to other modular systems also.

As presented in 4.3.2, the standard part of the modular system is central, thus the analyses in Figure 62 concentrate on that proportion of modular systems. Another possibility to compare modular systems is to compare the entire system while the platform between two systems can also include the variable proportion of the modular system. It is equally important to provide standardization as much as possible to the variable part of the modular system.

Commonality analysis

For product design, production and purchasing the establishment of generic product structure has been valuable for the case company. Entering a module, part or component code into the PDM system to figure out parent information for the specific module, part or component results in a long list of "where-used" information. The problem is the absence of knowledge which of the parents is valid in a case of configurable products. Also when configurable products are considered, the information related to what final product uses the module can be time consuming to define. Even if PDM systems include the information if, for example, a module is active or not, the issue for configurable products in changing environment is the maintenance of this knowledge. For the case company it took considerable amount of time to define active modules from the system since the knowledge related to validity of modules and parts was not up-to-date. This was mainly due to the changes that affected the product structure during its life cycle.

Configuration matrices have the knowledge of valid and generic product structures used at the moment. The main issue for configuration matrices for the case company is to provide exact information between the modules and the final product. By using the generic product structures valid modules, parts and components can be defined (Figure 63).

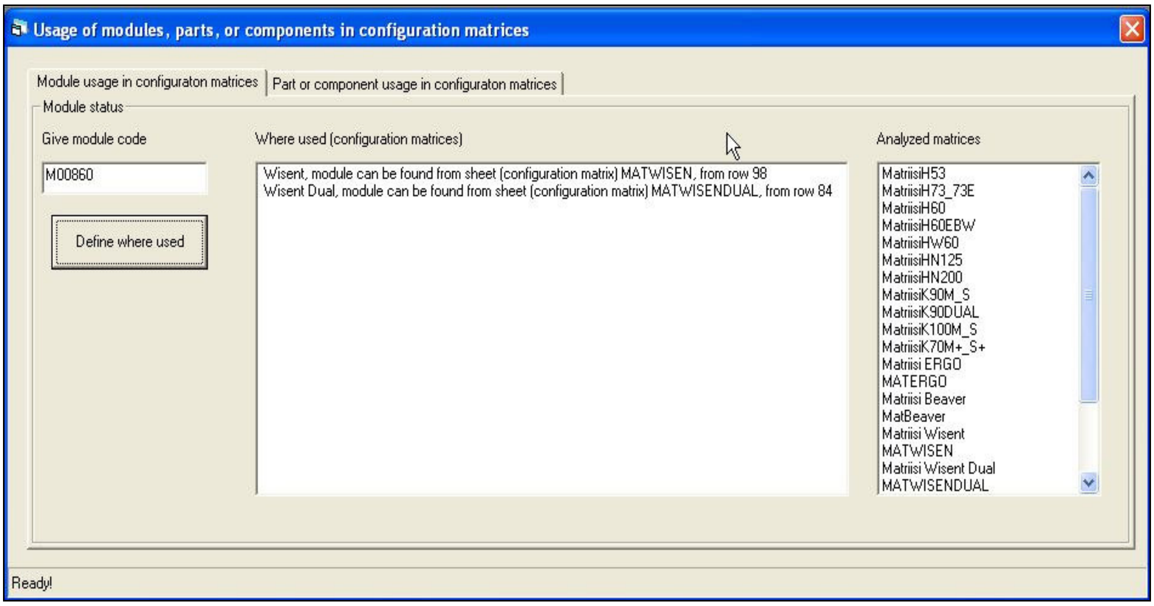


Figure 63. The parental information for modules, parts and components

For determining parental information for modules, there is no need to connect to the ERP or PDM databases since all the needed information is included in configuration matrices. The result of the analysis (Figure 63) is the configuration matrix or matrices where the module is used.

For determining the parental information for parts and components, the ERP or PDM databases need to be used in order to retrieve the specific BOM. For this application ERP databases were used. The idea is to use the generic product structure to determine modules and then to compare the part structure of the modules to the part or component needed to be analyzed. Thus, active modules of the generic product structure are used.

This type of analyses can be considered to be reverse configuration. This means that the generic product structures are used to define exact where-used information for the user. Thus, a part or a module is chosen and the configuration matrices can be used to trace the part or the module all the way to the option that will include the analyzed part or module to the configured product individual.

Cell re-engineering

One of the first applications for the case company was the cell re-engineering tool (Nummela 2003). This tool is used for production cells to determine valid parts and components and to separate the information from the ERP system so that invalid components can be identified from the databases as well as from the shelves of the company. During this research there were the following inconsistencies defined while using ERP system considering invalid part and components in production:

- Invalid parts and components found from the ERP system and from the shelves of the company
- Invalid parts and components found only from the shelves of the company
- Invalid parts and components found only from the ERP system

It is easy to get rid of the invalid parts and components found from the ERP system and from the shelves of the company by following the incoming change orders. More effective tools are needed for the other two cases presented above. Identifying parts and components to be either valid or invalid manually is time consuming, while defining invalid parts and components from the system alone is impossible manually. This section will introduce tools that can be used to define the presented inconsistencies.

Configuration matrices are used to define all the active modules, parts and components used for the specific cell. This analysis is well suited to a part of production where a group of configuration matrices can be matched with a production cell. When all the configuration matrices have been selected to be analyzed, the cell re-engineering proceeds to collect data (BOMs) from ERP databases for all the active modules presented in configuration matrices. After all the BOMs have been gathered the program will compare all the parts and components with the BOMs to define what modules, products and assembly location use the part or component. Part of a result from one of the analysis is shown in Table 13.

Table 13. The result of the production cell analysis

PART_ID	DESCRIPTION	IN MODULE	PRODUCT	ASSEMBLY LOCATION	WAREHOUSE	MACHINE INDEX	WEIGHT (kg)	USAGE LAST YEAR	USAGE THIS YEAR
0000067	SREW	M00070	HN125	Final assembly	06A9, 07B7	For multiple machines	0.01	3.00	5.00
		M00399	HN200	Preassembly					
0000354	BOLT	M01037	K90DUAL	Final assembly	06A5	For one machine only	0.10	33.00	222.00
		M00898	K90DUAL	Preassembly					
		M00901	K90DUAL	Final assembly					
0000967	CYLINDER	M00124	HN125	Final assembly	06A9, 07C9	For all the machines	11.00	122.00	421.00
		M00125	HN200	Preassembly					
		M00126	K90DUAL	Final assembly					
		M00127	K90DUAL	Preassembly					
		M00128	K90DUAL	Final assembly					

Table 13 shows the main information related to the analyses conducted in the crane assembly area. The result from the analysis can be used to determine the shelf locations as well as invalid parts from the warehouse. Again, the analysis is based on generic product structures and the information from ERP databases while the comparisons are done in the developed software system.

Next to the analysis there is always layout design involved in the process. Layout design and results from analysis provide a basis for effective iteration to come up with a suitable solution to the problem. One feature that has proven effective is the design for automatic warehouse system based on configuration matrices. When the parts and components are selected to be in the automatic warehouse system, the configuration matrices can be used to locate the part in respect to the module and finally to the feature and to its option that will include the part to the configured machine. Thus, the automatic warehouse can be optimized by arranging the parts according to their features.

After the layout has been designed and analysis done, the cell can be reorganized. While the number of parts in one analysis can be thousands for the case company, the analysis is complemented with a user interface in order to quickly give a certain part a status. The user interface is shown in Figure 64.

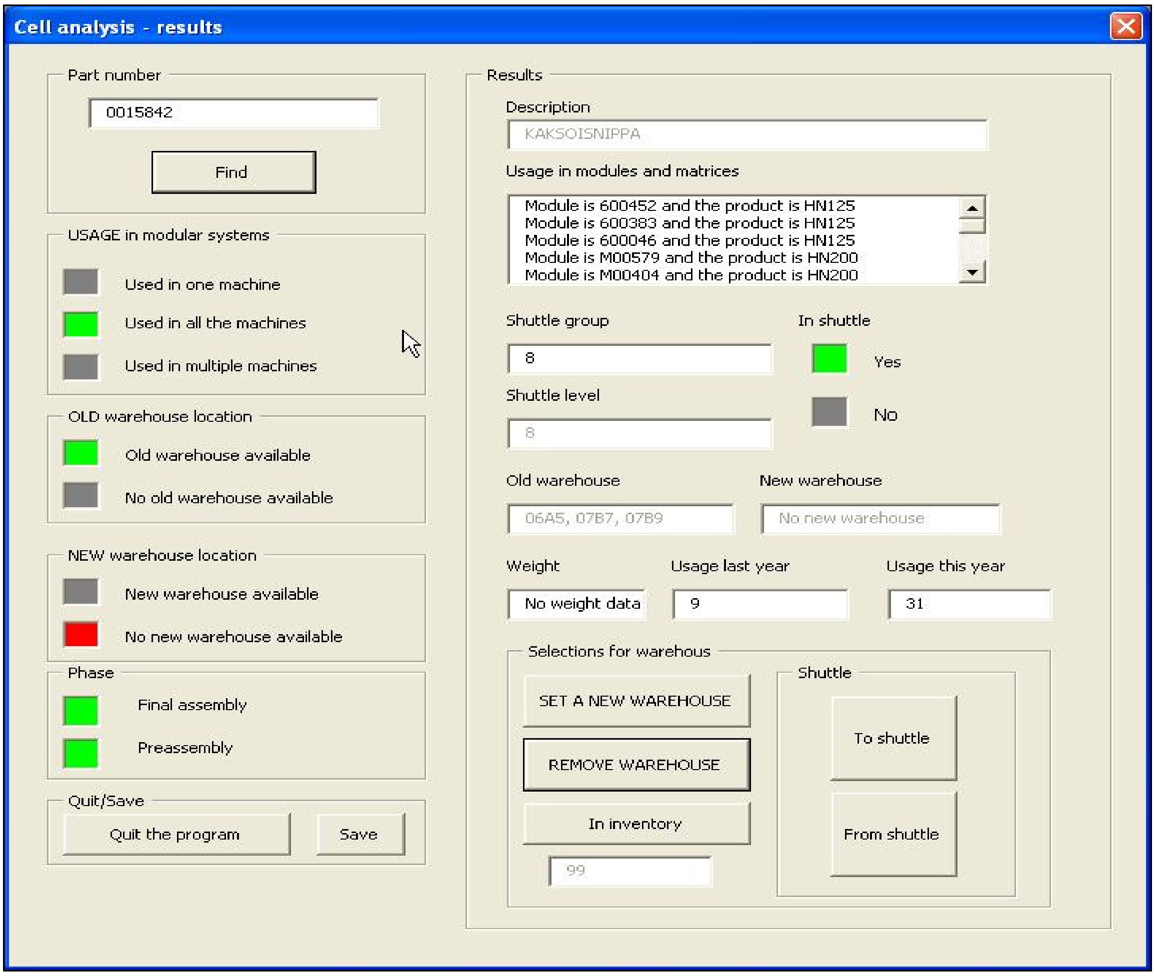


Figure 64. The interface to be used while re-engineering cells (Nummela 2003)

The idea of the interface presented in Figure 64 is to use the analysis made by comparing the active modules and their parts and components with the warehouse data from the ERP databases in order to interpret efficiently the status of a part or a component while re-organizing a production cell. During the reorganization of the cell all the parts and components are analyzed and new information updated by using the interface.

As seen in Figure 64, there is a field where module and the product can be evaluated. By using this data and the configuration matrices, it is possible to perform the reorganization by features. Also when establishing a new arrangement for the parts and components in the production cell, the usage in modular systems and phases provide effective means to locate parts and components. For the example the phase includes final and pre-assembly, thus the parts can be situated next to these locations first. Then by looking at the usage, the parts can be further arranged by the idea that they can be used in one model, all the models or some models.

Nummela (2003) provides the following results for the harvester head assembly area as presented in Table 14.

Table 14. The results from the harvester head cell re-engineering

	Total number of parts	Parts in shuttle	Invalid parts	Inventory value for invalid parts [Currency units]	Capacity of the cell (after cell re-engineering)
Before	915	137	137	43670	5
After	1082	205	110	11052	10
Change %	18.25	49.64	-19.71	-74.69	100.00

The change percent is calculated considering the before situation. As seen from Table 14 the reduction of 20 percent of the invalid parts produced reductions almost 80 percent of the total amount of the inventory value of invalid parts in the cell. The reason why all the invalid parts could not be removed was because these parts had reservations in the ERP system, thus the inventory balance could not be changed. These parts were only in the ERP system, i.e. waste that needs to be removed. The shuttle is an automatic warehouse system for the cell and all the parts in the system were rearranged by features and lots of space was freed. There is now almost 50 percent more parts in the system when compared to the before state. The reason why the total number of parts increased by almost 20 percent was the fact that not all the parts had inventory locations defined in the ERP system and new parts were introduced to the cell during the rearrangement. Finally, due to the layout redesign and cell re-engineering analysis the capacity of the cell was doubled.

Next to the above-mentioned benefits, the above example improved the quality of the information in the ERP system because all the parts and components were considered using the analysis. Also the product structures and configuration matrices were updated by the feedback from the cell re-engineering. For the case company the parts and components without inventory locations finally awakened the personnel to consider their processes more deeply. This, in turn, provided the case company to consider their lead times and defects from the upstream processes such as manufacturing. By doing this the rate of defects and the lack of parts from the upstream operations can be diminished when understanding the processes involved.

The analysis presented above was taken as part of the routine cell re-engineering of the case company. Cell re-engineering provided the case company to use a well defined process to provide enough information so that an entire cell producing configuration matrix based on product individuals could be rearranged and systematically improved.

Active module based analysis

As seen from the previous sections, all the analyses and tools are related to the generic product structures provided by the configuration matrices. Active module based analysis uses again the generic product structures and provides inputs mainly to manufacturing and purchasing. The user interface for active module based analysis is presented in Figure 65.

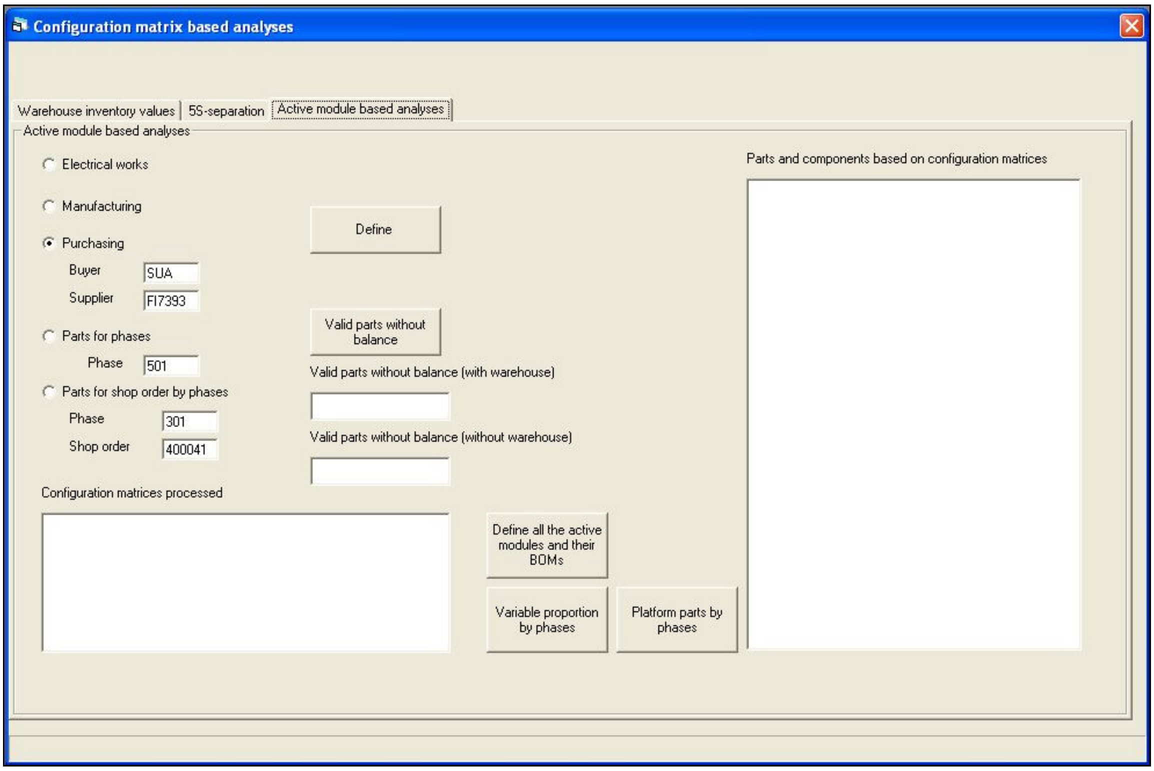


Figure 65. Active module based analysis

The main focus for active module based analysis is to provide exact information of the valid parts and components for:

- Electrical works
- Manufacturing
- Purchasing
- Assembly phases

For electrical works and manufacturing, the software system uses a list of active modules provided by the configuration matrices. By using ERP databases parts can be declared to be in either one of the production sites (electrical works or manufacturing). This is simply established by evaluating the parameters for the parts and components. For purchasing the only difference is that more parameters need to be included. The buyer and supplier are provided to produce exact information for the active parts and components for a wanted buyer and supplier. Finally parts for assembly phases require the user to insert a phase from the assembly system. By the list of active modules, BOMs from ERP and the given phase, the software system will establish all the active parts and components for the selected assembly phase. For example the parts, components as well as modules if existing for the third assembly line stage can be defined. Thus, the validity of the parts and components in the assembly site can be evaluated.

Even though it seems to be that when processes are working properly there is no need to consider validity of parts, components and modules from the line or anywhere else. In reality these types of problems are easily generated, and the most critical issue is that for companies using ERP systems these types of analyses are very hard to accomplish. The problem is that when the amount of data increases the validity of this data is increasingly questionable. As mentioned before, changes take place all the time and to check the validity of data can be very hard. As shown in the above discussion, the integration between configuration matrices and ERP or PDM databases provide a powerful tool to provide valid data from the systems.

Invalid parts analysis

Active module based analyses are based on the generic product structure and used to define invalid components in production (both in shelves and in ERP system as presented above), modules, parts and components for different phases, valid product mixes for different suppliers and valid product mixes for own production. These types of analyses are used to systematically and very efficiently satisfy the needs of many stakeholders of the company such as product development, production and purchasing without needing an extensive knowledge of configurable products or production system, as discussed above.

The invalid parts analysis uses all the possible configuration matrices and the ERP database warehouse knowledge to compare all the modules, parts and components found. Thus, the generic product structures and their parts and components are compared with the existing ERP database to figure out all the possible invalid modules, parts and components. The idea for this type of tool is based on 5S (Sustain, Standardize, Sort, Set in order, Shine) approach presented for example in Torvinen et al. (2004). The invalid parts analysis is related to the Sort phase of the 5S approach. In this phase all the invalid parts, components and modules are removed from the cell (Torvinen 2004). The idea for the invalid parts analysis is to use all the configuration matrices to provide an understanding of invalid parts from the entire assembly area. Cell re-engineering software system can be used to define invalid parts for a limited amount of configuration matrices.

For this type of analysis the need for additional information is critical since the change frequency and prototyping cause a situation where parts that are valid are not included in modules that are in configuration matrices, i.e. valid modules, parts and components can be declared invalid. The additional information gathered automatically at the end of the analysis from ERP databases used for the case company is shown in Figure 66.

Part	Description	Bacflush date	Event	Bacflush quantity	Arrival+ date	Event	Arrival+ quantity	Planner/byer
0062533	TELA	No events	No events	No events	9.12.2003 9:02	ARRIVAL	1	SPE
0062564	TELA	No events	No events	No events	22.12.2003 8:23	ARRIVAL	1	SPE
AS0011	POLTTOAINESHLI	12.2.2004 14:10	BACFLUSH	1	12.2.2004 9:42	ARRIVAL	1	LLE
P16762	PUSKURIKOTELO	12.2.2004 14:10	BACFLUSH	1	9.2.2004 6:12	ARRIVAL	2	LLE
P16755	PUSKURIKOTELO	12.2.2004 14:10	BACFLUSH	1	9.2.2004 6:12	ARRIVAL	2	LLE
P16463	TAKARUNKO	12.2.2004 14:10	BACFLUSH	1	29.12.2003 22:47	COUNT-IN	2	H05
0061906	HYDRAULIPUMPPU	12.2.2004 14:09	BACFLUSH	1	19.1.2004 12:03	ARRIVAL	7	TKA
0014241	O-RENGAS	12.2.2004 14:09	BACFLUSH	1	8.1.2004 12:14	ARRIVAL	500	SUA
0042097	SAE-LAIPAN KIINNIKE	12.2.2004 14:09	BACFLUSH	2	16.12.2003 17:44	COUNT-IN	180	TKA

Type code	Warehouse	MRP-code	Status	Where used in ERP
Purchased part	ULKO	A	A	, S64573False
Purchased part	ULKO	A	A	
Purchased part	02L2	A	A	, M00515True
Purchased part	01I1	A	A	, M00180False
Purchased part	01I1, 02A2	A	A	, M00180False
Purchased part	09A1	A	A	, M00184True
Purchased part	02K3	A	A	, M00186True
Purchased part	02K9	A	A	, 1200724False, 1200724False, 1200724False
Purchased part	02K1	A	B	, 1200724False, 1200724False,

Figure 66. Additional information used to determine invalid modules, parts and components

Different dates of events have been used to determine the usage and arrival of a certain module, part or component with the where-used information in order to determine if the module, part or component is valid or not. The Boolean expression behind the module number in the where-used information indicates if the module is found in the configuration matrices. If there is only the "false" expression, the module, part or component is most likely invalid.

When arranging the suggested list of invalid parts, manual interpretation is needed to accomplish a valid suggestion. The final outcome is usually a long list of parts and components that can be used by the warehouse personnel to systematically approach the problem of invalid parts and components. This software tool compares parts and components so that all the issues presented in cell re-engineering (above) are revealed. Thus, also the invalid parts hanging in the ERP system can be eliminated. For the case company the invalid parts list usually equals around 6…8 percent of the entire inventory value of the company. This is a considerable amount of waste in the system.

This analysis also provides a counter part for the invalid parts elimination. When a part, component or module is declared invalid and in reality it is not, there is a problem with the configuration matrices or module BOMs. This can now be used as feedback to update the configuration models or the BOMs having the problem.

Product kill analysis

The product kill analysis tool is meant to be used at the end of the product life cycle, i.e. when the product is experiencing a ramp-down phase in production. Modules, parts and components need to be analyzed to decide what modules, parts and components are invalid and can be removed from the warehouses during the ramp-down phase. Invalid modules, parts and components can then be relocated to be used in after sales operations as spare parts. The question is how the invalid parts, components and modules can be defined. For the case company this phase was usually done with the personnel in assembly by marking parts and components that are not used. This procedure was time consuming and the result can be far from satisfying.

With the product kill tool, production development can run analysis that takes only few minutes to determine invalid modules, parts and components. The user interface is shown in Figure 67 for product kill tool.

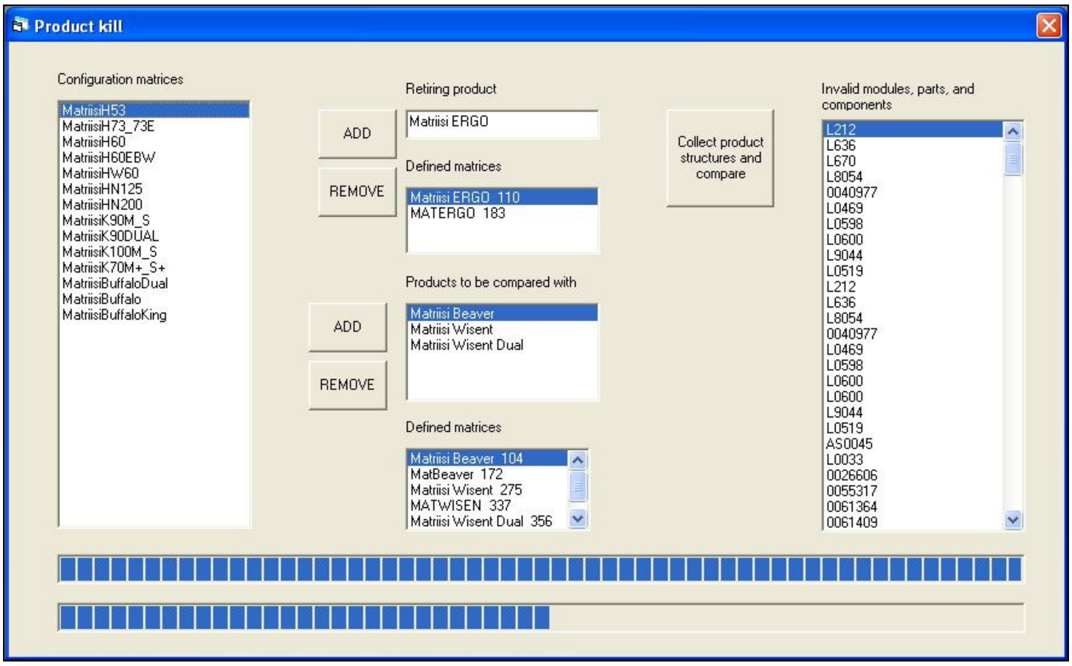


Figure 67. User interface for the product kill analysis

The product kill analysis is based on configuration matrices and their generic product structures. In order to figure out the level beneath the module level, ERP or PDM integration is needed. When running the analysis the user defines the retiring product and the products that are to be compared with the retiring product. The result of invalid modules, parts and components is listed into the list box shown in Figure 67. The result includes all the invalid modules, parts and components from the retiring configuration matrix, i.e. these modules, parts and components are only used in this product. The invalid modules, parts and components only in the ERP system (see cell re-engineering above) cannot be defined and the invalid part analysis is used for this purpose.

The product kill analysis compares different matrices and the BOMs of the modules to deliver a solution for the ramp-down phase. When a product is withdrawn from production, the matrix presenting this product is compared with all the other matrices. By doing this, the result will provide all the parts that will become invalid when the product is taken away from production.

APPENDIX 4

The comparison between different configurators available

The conventional configurators are mainly established by a manual configuration rules generator. This module is used to insert the configuration rules into the system. There are various ways that this type of task is handled in the conventional configurators. This module can handle functions and various different kinds of rules. Thus, more than "AND" and "OR" operators are used.

Next to these configurators it needs to be noted that for the built in-house configurator the K- and V-matrix (Bongulielmi 2004) based configurator provides the best benchmark. Configuration matrices presented in this research are a subset of K- and V-matrices. These matrices can be used even more effectively in product development than configuration matrices established in this thesis. The differences between the developments of these two types of matrices lie in the fact that configuration matrices presented in this thesis are considering the importance of configuration knowledge more broadly including also the production system heavily next to modularity. This means that the configuration knowledge is not only considered in the context of the configuration task and related processes. Thus, many aiding tools have been established next to the configurator.

The benefits of the configuration matrices and the developed configuration knowledge generator reveal the advances in the context of other available configurators. In the following table (Table 15), the differences between the conventional, K- and V-matrix-based configurator and the built in-house configurator are presented.

Table 15. The differences between available configurators

	In-house built configurator	Manual rules generator based configurators	K- and V-matrix based configurator
Automatic rules generator	x		x
Automatic basis for pricing	x		
Change process algorithms to define the validity of configuration knowledge for product changes	x		
Automatic reconfiguration of order books	x		
Automatic re-establishment of configuration knowledge	x		x
Revisioned configuration knowledge	x		
Easy reconfiguration process	x		
Customer change process using the validity algorithms	x		
Easy formation of matrices	Only in Excel	Not applicable	Well defined interface

The main goal for the in-house built configurator has been to establish a workable configurator than supports the processes even when products and their specifications change. It has not been enough to provide a stand alone configurator that works in stable conditions.

Changes will occur and this is the fact that has been concentrated on. The similarities between the established configurator and the K- and V-matrix-based configurator in Table 15 are the automatic rules generator and the re-establishment of the rules after configuration knowledge has been changed. This is also the main benefit when conventional configurators are considered. The main benefit that the K-and V-matrix has is the well established interface for creating the matrices.

The benefits related to established configuration matrices and the configuration knowledge generator (presented in section 3.8) are related to the change management of the configuration knowledge. While the automatic rules generator is the core for the entire database creation, the process includes many automatic features such as basis for pricing and revisioning the configuration knowledge. By revisioning this knowledge the algorithms used to validate the configuration knowledge become useable. Now the various benefits that other configurators lack can be introduced. These systematic processes now enable the company to use the configurator efficiently while changes take place during the life cycle of the products and order. While after sales processes are very important, the revisioned configuration knowledge provides an easy process for reconfiguration. This means that old configuration knowledge can be used to reconfigure a product or a product can be updated by reconfiguring the product with new options introduced by the latest configuration knowledge. One issue to be dealt with is the integration between the ERP and PDM systems. The main integration is done so that the configurator and the related tools use ERP and PDM databases to read information. Integration in the other way is only needed to provide the configured product individuals from the configurator to ERP or to PDM system. This is very critical since the automatic update of the order books for revisions and version changes for modules is dependent on this feature.

When considering the main benefits for the configurator presented, the effects are considerable. The established configurator includes the main features provided by the configuration matrices. As the configuration matrices are the core for the entire approach and the main benefit is the configurator, another benefit related to configuration matrices are the tools established to provide systematic processes next to the configuration process. These processes and tools use the configuration knowledge related to the configurable product presented in configuration matrices.

APPENDIX 5

The effects of modular product architecture

Modular product architectures have many implications to the business processes of the company. Mostly the implications are positive, but also negative effects can be found. Stake (1999) considers the benefits of the modular product architecture to be the reasons for modularity. According to Ulrich (1995), the product architecture has major effects on five managerial issues. These five major effects are (Ulrich 1995):

- Product change
- Product variety
- Component standardization
- Product performance
- Product development management

The ease of product change can be far greater in modular architectures than in integral ones. The product variety and component standardization are connected to the idea of commonality and distinctiveness, i.e. the purpose is to generate customer variants and at the same time to try to limit the complexity and cost by creating enough commonality between the products in the product family and also between the product families (Ulrich 1995).

Ulrich (1995) divides the product performance to local performance characteristics (physical properties of a local region of the product) and global performance characteristics (physical properties of most components of the product). Modular product architecture can optimize the local characteristics, but can sub-optimize the global ones. Product development management is also affected by modular architectures. The main difference is the concentration on performance optimization of the product versus the optimization of the product considering the organization. Product development management is also affected by the formation of the product architecture, by the division of the modules to the applicable teams and also by the possibility of concurrent engineering and testing (Ulrich 1995).

According to Dahmus et al. (2001), ideal product architecture decomposes the product into useable and meaningful modules. These modules are intended to serve the organizations as well as possible in order to create success and save cost. According to Erixon et al. (1994), modularity has the following advantages considering the entire company:

- Concurrent manufacturing of modules shortens the lead time in manufacturing, especially in assembly
- Concurrent engineering shortens the lead times of product design process
- Stock levels decrease due to the shortened lead times of the production system and due to the decrease in the need of stock management
- Material costs decrease due to the decrease in number of parts
- The quality is secured with module specific drawings and with testing the modules before the final assembly

- The routines for quotation, design and configuration of customer specific structures can be made more efficient
- Maintenance and upgrades become simpler because of the standardized interfaces
- The development of the production system as well as the product become easier because the future goals can be divided into the incremental development steps considering the modules

One of the main impacts of modularity is the explosion of the product variants. The principles of mass customization are usually based on the idea of modularity (Pine 1993). According to Baldwin and Clark (1997), the customers can mix and match elements to come up with the wanted variant that suite the customer need. They also bring out the possibility to manufacture in different sites and then bring the subassemblies together to enable an effective final assembly. Stake (1999) considers the explosion of variants and the problem that it causes to the production to be solved by the commonality modularity offers.

While considering the organization the modularity should be designed into the product considering all the stakeholders. According to Lapinleimu (2000), modularity of the product has the following implications to the different parts of the organization:

- Marketing
    - configurable product
- Product design
    - only part of the modules needs to be changed during the product generation change
    - Time-to-market gets shorter
- Material procurement
    - clear entities to purchase
    - modularity guides the formation of supply chain and partnership between the suppliers
    - number of manageable parts decreases
- Assembly
    - concurrent assembly shortens the lead time
    - the WIP (Work In Progress) of the assembly decreases
    - the final assembly layout becomes clearer
    - module testing decreases the need for repairs after the final assembly
- After sales
    - changing the modules fastens the maintenance and repair of the products

Lapinleimu (2000) sees the aspects of material procurement and assembly to be similar in the context of the product structure. The purchase department needs clearly defined parts, components, and assemblies in order to function. As the assembly structure and operations management are considered the assembly needs clearly defined assembly structures while marketing and design departments need functional structures.

Pahl and Beitz (1986) consider the implications of a modular system to the customers to be as follows:

- Short delivery times
- Better maintenance possibilities
- Better response from spare parts
- Subsequent functional improvements and enlargements can be done within the variant space of the product
- The possible sources of error are nearly eliminated because of the well defined industrial design

Next to the ideas of Pahl and Beitz (1986) one of the main reasons and benefits of the modularity is the above mentioned explosion of the variants experienced by the customer (Baldwin and Clark (1997), Pine (1993)).

The benefits related to modularization seem to be product development based considerations. Short delivery times appear, but the way these shorter delivery times can be achieved is not considered. The order-delivery process can be considered to include the following phases:

- Definition of the customer specification
- Confirmation of the order
- Creation of the customer specific product structure
- Component purchases
- Manufacturing
- Final assembly
- Testing
- Delivery

While modularization simplifies the use of the product configurator (see section 2.3), the three first phases can be made more efficient, i.e. considerable amount of time can be taken away from the processes. For the purchasing, manufacturing and assembly operations the modularity provides means to detach the product structure from the conventional marshalling methods (e.g. MRP 'Materials Requirement Planning' and push environment). Thus, modules can be assembled before hand making the manufacturing of parts also detached from the push system. This means that quality aspects are improved and the delivery time that the customer experiences is drastically reduced. Finally, the testing phase can be considered testing the final assembly since all the pre-assembled modules can be tested prior to the final assembly. All of the above mentioned issues are also related to delayed differentiation (e.g. Ulrich and Eppinger 2000) while the mechanisms of doing this is also considered. While modularizing product structure there are possibilities to reach the similarity between production system and product structure in many levels. Thus, the point of differentiation in production can be defined when considering the standard proportion and the varied proportion of the product structure.

The product consists of modules, sub-modules, kits (set of parts) and raw materials (Lapinleimu 2000). Considering this categorization the modules provide the distinctiveness required by the customer while the level of standardization can be taken furthest in the raw materials. This is also related to the companies' strategies e.g. what is the competitive know-how for the company, thus what parts of the production can possibly be outsourced. Also the deviation experienced by the production system is related to the above categorization. The more standard work there is in the upstream operations the less deviation is experienced by the final assembly. Thus, when sub-modules are assembled from standard parts and components and final assembly uses standard sub-modules to provide variants for the customer, the system can be operated consistently. While continuously reducing the time required for the order-delivery process, the ability to compete in the markets also improves when the company can deliver customers specific product individuals quicker than the competitors can. The reduced time is also related to the deviation experienced by the company producing modular products while also concentrating on the core business is beneficial. Not all the time can be diminished from the order-delivery process, but the point is that the most of it can be reduced to appropriate levels.

Even if there are numerous benefits when using modular architectures some drawbacks can be found. According to Pine (1993), the drawbacks are as follows:

- The performance of a product can be optimized and its manufacturing costs lowered by reducing or eliminating modularity
    - only applicable with single product
    - the power of modularization in lowering costs is greater when number of similar but clearly differentiated products are manufactured
- Customers can identify some parts of the product to be too similar
    - the main thing for designers is to find what is most personal for the customer and differentiate it
- Competitors can reverse-engineer modular designs more easily
- Less innovative solutions can occur over time

Ulrich (1995) adds to the above list the idea that modular product structures are not as well suited for optimizing performance features like mass, acceleration and size.

Considering the possibility to have an easily configurable product structure and the similarity between the product structure and production simultaneously creates major benefits for the company considering cost, configuration management, operations management, purchasing and product design. Also when the product structure can be used to analyze the production system to handle the life cycle impacts systematically, the maintainability of the product structure as well as production system increases.