Interconnection Based Product and Process Documentation

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens het besluit het College voor Promoties
in het openbaar te verdedigen
op donderdag 13 maart 2008 om 15.00 uur

door

Marco Groll
geboren op 20 april 1967
te Eislingen/Fils

Dit proefschrift is goedgekeurd door de promotor
Prof. dr. ir. F.J.A.M. van Houten

Interconnection Based Product and Process Documentation

Marco Groll

Interconnection Based Product and Process Documentation
Ph.D. Thesis
ISBN 978-90-365-2630-2
Marco.Groll@{daimler.com,web.de}

To Maximilian, Yannick, and Ann-Kathrin

Preface

The present dissertation has emerged during the many years of my engagement as employee in the units "Global Research and Development" and "Daimler Trucks" at Daimler AG. The intense collaboration in numerous national and international projects in the context of product and process documentation had a substantial influence on this work and ensured a close link to practical work in companies. In the course of this study, my conviction that the method of an interconnection based product and process documentation which is presented here is the sole alternative to the currently known documentation methods which have reached their limits long ago was shaped ever more firmly. Not least was my conviction assured in the numerous, sometimes vivid discussions with critics and "opponents" of this method. I would like to express my gratitude to them.

My special thanks go to Mr. Prof. Dr. ir Fred van Houten from Twente University who accompanied this study with greatest interest. His wealth of experience and his very pronounced analytic thinking made him a particularly helpful partner in our numerous discussions. Notwithstanding the great distance between Enschede and Ulm/Stuttgart, his remarkable patience and the trust he placed in me helped me to keep my spirits high even in critical phases of my work. With his support, he made a valuable contribution to the success of this study. My thanks and appreciation also go to the members of the examination board Prof. Dr. F. Eising, Prof. Dr. ir. J.I.M. Halman, Prof. Dr. R.J. Wieringa, Prof. Dr. ir. J.C. Wortmann, Prof. Dr. ir. A.C. Brombacher, Prof. Dr. N.D. Du Preez, and Dr. S. Bhagavathula.

I would like to thank Dr. Seshu Bhagavathula, Alfred Katzenbach, Dr. Siegmar Hassis, and Klaus Schott for providing the framework conditions on behalf of Daimler AG to compile this dissertation.

I express my special gratitude to Eckhard Monke, the "discoverer" of interconnections, for the many interesting and fruitful discussions and friendly cooperation. In this context, I also thank Dr. Rainer Ostermayer, Udo Kröger, Holger Iloge, Marc Hofmeister, and Torsten Cordes.

My thanks also go to Stefan Römer and Axel Braun for their support in creating the IT prototype.

I owe special thanks to Ingrid Sonnleitner from sonntrans for translating the dissertation into English.

I pay tribute to Preeti Bedi who accompanied me during the final phase of my dissertation. Her commitment and her appreciation for my person and work helped me to see the essential things of life on a distinctly broader context.

I owe special thanks and deepest appreciation to my parents Alfred and Ingeborg Groll for having supported me throughout my life and in the various phases of my research work. Their love and readiness to be always at my side in difficult situations have given me the trust and energy to succeed in solving challenging tasks.

My endless gratefulness and admiration go to my children Maximilian, Yannick, and Ann-Kathrin and my beloved wife Sabine. It is to their endless patience that I could carry out this work successfully. Their love and readiness to pass on many hours of being together gave me the motivation to continue even in moments of hopelessness and led this dissertation to a successful end.

Ottenbach, January 2008

Marco Groll

Abstract

The globalization of markets has had distinct effects on industrial manufacturing companies during the past years. The design of processes is gaining significance in this context becoming a critical factor for the long-term success of a manufacturing company. Information as production factor with increasing meaning has developed into an important success factor. The method of product and process documentation is of special importance. Deeply embedded into the technical workflow organization, the product structure maps all product variants to form the basis for any process activity. This is particularly true for customer order related series manufacturers with large variant diversity as it is reality in automotive industry.

The method of product and process documentation based on interconnections which is developed in this work considers the central significance of the product structure and provides for the prerequisite to create flexible and continuous business processes. The interconnections documentation starts from the methods of product and process documentation used in automotive industry today. It can also be used in other manufacturing industries, all the more when the companies are confronted with the increasing problem of growing product and process complexity.

This study on interconnections documentation is divided into four main parts:

The first main part of the work addresses the current situation of manufacturing companies using automotive industry as an example. The problem of variance and complex processes prevailing in the companies in relation with the information systems will be reviewed in detail to reveal the special significance the product and process documentation methods have in this context (see chapter 1). In the further course of this main part, current documentation methods will be analyzed (see chapter 2). The analysis will be completed with the investigation of selected implementations of integrated product and process documentation.

Based on the analysis results of the first part, the second main part formulates the requirements placed on an interconnection based product and process documentation (see chapter 3). Following the analysis phase, the requirements are formulated as seen from the difficulties resulting from variants and process complexity.

The third main part of the dissertation describes the concept of the interconnection based product and process documentation (see chapter 4). At first, the methodological approach of this documentation type is sketched in an overview. This is followed by details about the

underlying concepts of the interconnections documentation. To start with, the methodological procedure is addressed followed by a few application examples (amongst others the integrated change management). The third part closes with a description of a step-by-step migration concept.

The fourth and last main part describes the design and prototype implementation of the interconnection based product and process documentation. At first, the object oriented design of the interconnections documentation considering the Unified Modeling Language is introduced (see chapter 5). This is followed by an architectural concept and the relevant technological basics for a prototype implementation. Using the business scenario "ITego" as an example, the methodological procedure of the interconnections documentation will be presented (see chapter 6). The dissertation on interconnections documentation closes with a summary and outlook for the further procedure (see chapter 7).

"*The whole is more than the sum of its parts.*"

Assigned to Aristotle (384-322 BC)

Contents

Part III CONCEPTUAL LAYOUT

Part I

INTRODUCTION

1 Introduction

In the past years, the requirements placed on industrial manufacturing companies have increased enormously due to ever changing market situations (cf. [Trippn02]). Companies have to react towards both external and internal changes (cf. [Thaler01]). The external pressure for modifications is triggered by

- increasing globalization
- increasing market orientation
- growing model variance
- increased quantities
- shorter product cycles
- decreasing target costs

(cf. [Schött99]). The internal pressure for modifications, in turn, mainly grows due to

- growing product complexity
- increasing modification frequency of parts

The process design is gaining in significance. In the ever tighter global competition, it is critical for the long-term success of a manufacturing company. Processes are dynamic and call for an adaptation to the changing external and internal requirements. The aim are flexible and continuous processes which support the entire product lifecycle from product planning and generation in engineering, procurement and production to distribution and service (cf. [SapPlm01]). In this context, information as production factor with increasing meaning has developed into an important success factor (cf. [Iloge01]).

With a growing cost pressure, the readiness of many manufacturing companies rises to invest considerable sums in expanding their information technology (cf. [Schött99]). With the introduction of PPS or ERP systems respectively in the areas relevant for planning and production, a growth in productivity has already been achieved recently. Likewise, an increase in engineering efficiency can also be observed in the areas preceding manufacturing with the introduction of EDM/PDM[1] systems (cf. [Eigner01]). Nevertheless, the originally

1 In this work, the terms EDM and PDM are used as synonyms.

announced targets regarding increased productivity and improved engineering efficiency have not been achieved yet.

Practical work life shows that a quantum leap cannot be achieved with the use of latest information technology alone. A continuous method for a complete description of products and processes to control data administration and provide data in a flow-controlled way across tasks without media breaks in companies or corporate groups is missing (cf. [Schött99]).

In this context, *Technical Documentation* with the underlying documentation method is of special importance (cf. [Trippn02]). Deeply embedded into the technical workflow organization, the product structure maps all product variants to form the basis for any process activity (cf. [Eigner01]). This is particularly true for manufacturing companies with large variance and customer order related manufacturing as it occurs in the automotive industry, for example (cf. [Ohl00]).

The method of product and process documentation based on interconnections[1] which is presented in this work considers the central significance of the product structure and provides for the prerequisite to create flexible and continuous business processes. The development of the interconnections documentation starts from the methods of product and process documentation as they are used, amongst others, in automotive industry today. This accommodates the requirements and experience of practical worklife. The interconnections documentation, however, is by no means restricted to the aspects of automotive industry. The methodological approach described here can also be transferred to other manufacturing industries, in particular if they are confronted to an increasing extent with the problem of a permanently rising product and process complexity facing global competition.

1.1 Situation of automotive industry

In the past years, the globalization of markets also has substantially affected automotive industry (cf. [Iloge01]). International networks consisting of OEMs[2], system and parts suppliers, engineering and sales partners (cf. Fig. 1.1) call for a global organization and coordination[3] (cf. [SapPlm01]). The international presence and the increased transparency of cost differences also mean international competition for the vehicle manufacturers and their suppliers (cf. [Berg93]). According to [Ohl00], the resulting competitive situation leads to an unconditional cut-throat competition between the suppliers.

1 Referred to as "interconnections documentation" in the following.

2 Manufacturers which use original products of other manufacturers for their own products in mutual agreement to launch the finished products with their own label.

3 See requirement 3.2.1.

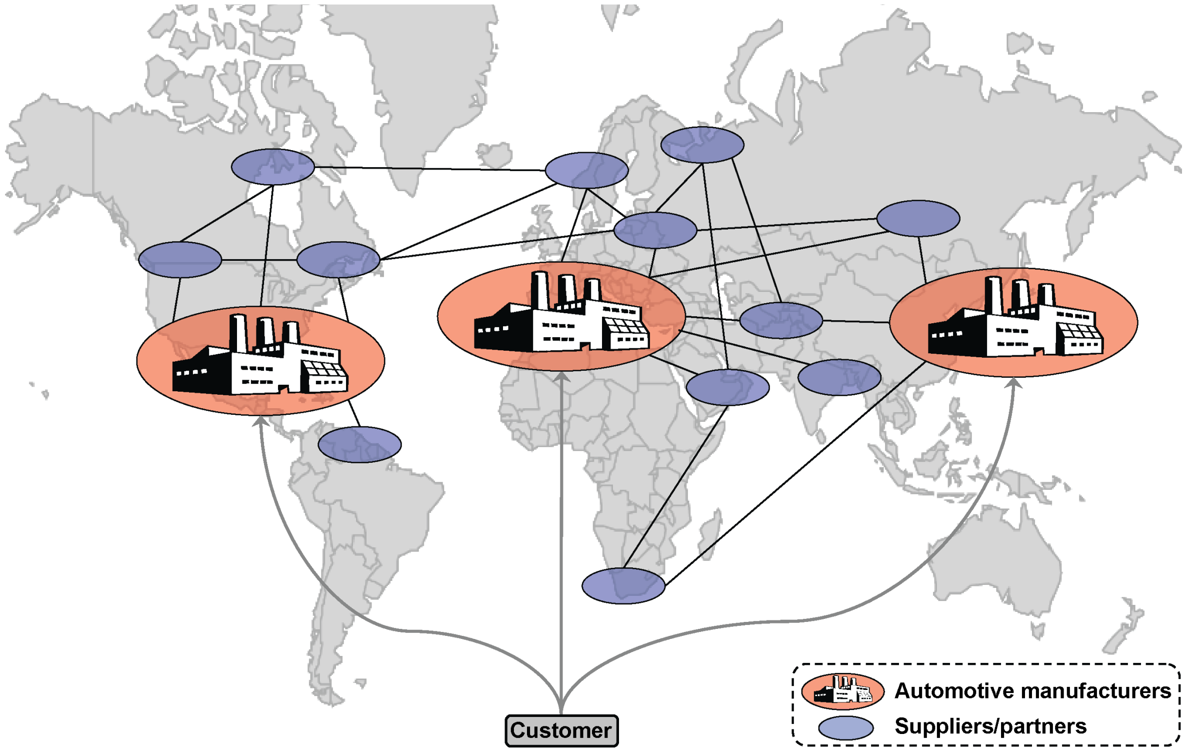


Fig. 1.1: Automotive industry network to be organized globally

Designing and optimizing corporate-wide business processes as well as processes across companies is becoming ever more important[1] (cf. [Thaler01]). Vehicles have to be developed faster at lower costs while increasing customer benefit. This is added by the fact that the vehicles have to comply with a variety of partially differing legal regulations in an international market.

In global competition, the automotive companies strive to offer a well balanced and complete product portfolio, from minicars to luxury coaches. Hence, model and variant diversity increases permanently. Apart from variant diversity, the development of new innovative vehicle concepts also results in an increased product complexity[2].

Automotive industry is characterized by an increasing variant diversity and increasing product and process complexity at the same time. The methods, workflows and IT systems required for engineering and production become ever more complex (cf. [Trippn02]). The following sections of this chapter will examine the situation of the automotive industry focusing on

- variant diversity
- process complexity
- the interaction with information systems

1 See requirement 3.2.2.

2 See requirement 3.1.4.

1.1.1 Variant diversity

According to [Warnec93], there are two general approaches to consider the problem of variance:

- avoid variant diversity
- manage variant diversity (from technical and organizational viewpoint)

Since long, automotive industry has considered variant diversity to be the strategy of the future and tries to use it as competitive advantage[1] (cf. [Thieße96]). It was not always like this which is revealed in Henry Ford's famous phrase "*You can paint it any color, so long as it's black*" [Merced95a]. Since the related concept of a product portfolio with extremely few variants, model and variant diversity has almost exploded (cf. [AutKol90], [Meinin94]). Alfred Sloan, President of General Motors, discovered the advantages of a product portfolio rich in variants as early as at the beginning of the 20s. In contrast to Ford, Sloan realized that the different customer wishes could be fulfilled better if several vehicle variants were offered (cf. [Ohl00]). With the demand for a better customer orientation which has grown from this time on, the individual customer needs have to be considered to an ever rising extent. Meanwhile, customers can select that vehicle which lives up to their personal and individual demands free of any compromise. From a variety of special equipment, most diversified colors, different engines and other services, customers can configure their personal vehicle today (cf. [Wahl95]). Most recently, the term "profitable mass customization" has been associated with the target of developing and producing customized products in a fast and economic way (cf. [Piller03]).

How the fulfillment of individual customer requirements based on freely selectable vehicle characteristics affects variant diversity is explained in the example of the Mercedes-Benz Roadster SLK (cf. [Merced95a]) by [Ohl00][2] (see Fig. 1.2).

The original Roadster concept was based on few variants only. Two engine variants and 26 freely selectable optional equipment variants to be combined with each other in any combination[3] (cf. [Merced96a]) result in as many as 2^{27}, i.e. more than 134 million possible vehicle variants.[4] With an assumed annual production of 42,000 vehicles (cf. [Merced96b]), it would be possible to only manufacture a mere 0.031% of all possible variants in a year[5]. To actually produce any possible variant at least once, production would have to run 3,200 years.

1 This mainly applies to the European and American automotive industry. Japanese vehicle manufacturers, however, decided to avoid variant diversity. In contrast to the widespread opinion, Japanese car manufacturers produce fewer variants than assumed (cf. [Womack91]).

2 [Hübner75] described a similar example for the VW Beetle as early as in the 70s.

3 Ignoring paint and upholstery.

4 In CVD, 10^{100} (> 2^{300}) theoretical variants are said to exist for busses (cf. [Martin88]).

5 If customers do not order identical vehicles.

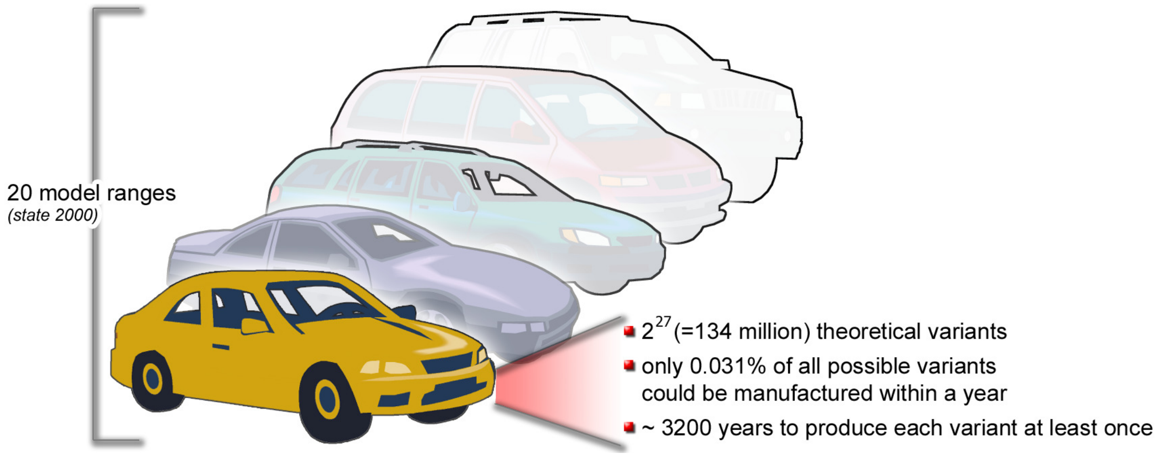


Fig. 1.2: Variant diversity

According to [Herlyn90], the largest portion of all possible vehicle variants is never produced, and many vehicles are produced only once throughout the entire product lifecycle (cf. [Bastia94a], [Bastia94b]). In the seventies, a repetition frequency[1] of 1.6 was determined for the VW Passat (cf. [Briest93]). Various investigations in the automotive industry revealed that today, the repetition frequency value often is 1 (cf. [Ohl00]). There almost is not a single vehicle of a German car manufacturer that is identical to any other vehicle (cf. [Bliese92], [Lingna94]).

The anonymous mass product has almost completely disappeared from the market, and the changeover from a market of offers to a customer market has almost been completed (cf. [Ohl00]). According to [Schuh89], variant diversity has risen disproportionately to product quantity over the time. [Pfeiff89] considers it to be the central problem of corporate management.

In this context, the method of product structuring and documentation is the core piece of any company (cf. [Brändl95]). Apart from a complete mapping of all product variants[2], the product structure also has to describe the structural associativity of the individual components/parts of a product variant among each other[3] (cf. [Eversh93], sim. [Rathno93]). Hence, the product structure expresses product complexity. Complexity is described by element variance (type and number of elements) and the relation variance (type and number of relations) (cf. [Steinm99]). The increasing use of electric and electronic components does not only increase the number of vehicle variants, but also the complexity of the vehicles themselves. It is less the number of different parts in a vehicle than their interrelations which is of importance. More than ever, product structures must be capable of mapping webs which consist of parts and their interrelations[4].

1 The ratio of produced quantity to number of variants.

2 In the following, the number of all theoretically possible product variants shall be referred to as "first level completeness".

3 See requirement 3.1.1; 3.1.4.

4 See requirement 3.1.1.

However, the methods of product structuring applied today are limited to describing relations of the type "consists of" or "is used by" (cf. [Eigner01]). In this context, there are a few weaknesses. While the current methods of product structuring primarily strive for a (vertical) continuity from the entire vehicle (as seen from Sales) down to the parts level (as seen from Engineering), a (horizontal) integration of process information[1] is only possible to a limited extent[2,3]. Henry Ford recognized the immediate effects of variant diversity on process complexity early when saying "*We believe ... that no factory is large enough to make two kinds of products.*" (cf. [Rathno93]). According to [Lingna94], variant diversity of the manufacturing companies results in an increasingly heterogeneous production program.

1.1.2 Process complexity

Apart from the historically rather young development of a globally organized value adding and the related international competition (cf. [Iloge01]), it has always been the aim of automotive manufacturers to optimize the entire product lifecycle regarding cost, time and quality[4] (cf. [SapPlm01]).

The consequences are increased demands in the continuity of the processes and flexibility of the organizational structures. Although quality and productivity have distinctly improved by the use of information and communications technologies during the part years (cf. [Iloge01]), the awareness of the interrelations of processes in their entity was pushed to the background[5]. In practice, the totality of the processes is rather split into partial processes. As a characteristic for the automotive industry, the processes are split into the core processes "customer order processing" and "product generation" (cf. Fig. 1.3).

The splitting of processes into separate technical units often leads to a variety of organizational interfaces resulting in a heterogeneous system world[6] (cf. [Thaler01]). Processes end artificially at system and department boundaries. The functional splitting into areas with the trend of suboptimizing within the respective areas increases friction and information losses (cf. [Wienda95]). The existing functional organizations are not sufficient to maintain competitiveness on the long term. It is not possible to quickly react to market and customer requirements - or to a limited extent only (cf. [SapPlm01]).

1 Procedure data, for example, as seen from work planning (cf. section 2.5.2).

2 In the following, the complete documentation of product, process, and resource information of a product shall be referred to as "second level completeness".

3 See requirement 3.1.1; 3.1.2.

4 See requirement 3.2.2.

5 See requirement 3.2.2.

6 See requirement 3.1.4; 3.2.2.

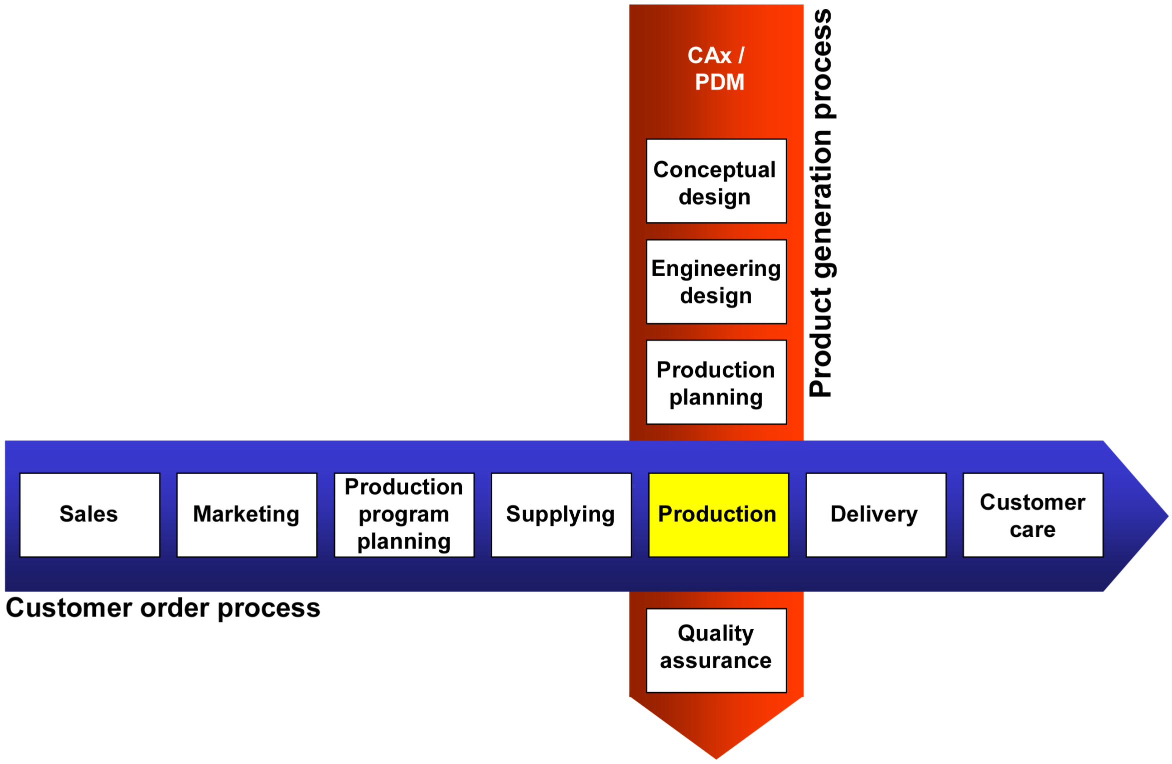


Fig. 1.3: Core processes of customer order processing and product engineering (Source [Groll99])

According to [Bullin96], a changeover from a functional to a process-oriented corporate organization is taking place in automotive industry. As basis for all activities of the various business units, the product structure lays the foundation stone for designing flexible and efficient processes.

The product structuring method forms the basis for mapping all product variants. For [Eigner01], it is the linking element between product and process[1]. It is considered a central means of product documentation (cf. [Ohl00]) and serves as backbone along the entire product lifecycle (cf. [Brändl95]). Typically, the product structure is created during product engineering. Other areas, like for example sales and marketing, work planning, calculation, procurement, production and maintenance use the data parallelly or downstream[2]. A "wrong or insufficient" product structure causes direct problems in the downstream processes.

A product structure defined by the product documentation serves, amongst other aspects, as link between the numerous digital views and the sole physical representation of an individual customer vehicle[3]. The degree of supporting a seamless transition from digital product engineering[4] to the physical world of vehicle manufacturing can be interpreted as criterion for an ideal product structuring method.

1 See requirement 3.1.1.

2 See requirement 3.1.2; 3.2.3.

3 See requirement 3.1.6.

4 See requirement 3.2.3.

According to [Katzen02], product structure has turned into a decisive competitive factor and represents the most valuable asset of an automotive company.

The designing of flexible and continuous business processes is in direct associativity with the method of product structuring and product documentation (cf. [Schött99]). The method of *Technical Documentation* forms the basis for a successful instrument of process control inside and across companies[1].

1.1.3 Information systems along the core processes

The splitting into the two core processes "customer order processing" and "product generation" also results in a separation into commercial and technical information systems.

The basic data consisting of master data[2] and structural data[3] (cf. [Eigner01]) form the nucleus of the respective system worlds. The core processes always must be provided with the current data. Apart from the product structure, work plans and capacity data belong to the key basic data according to [Zäpfel91].

Commercial information systems like PPS or ERP systems mainly support the customer order process (COP) as far as the business management and planning functions are concerned. Typically, these systems support the organizational flow in the areas of human resources, financing and accounting, controlling, material planning, production planning, sales and administration. However, they are not developed with the objective to integrate the various manufacturing systems (cf. [Schött99], [Thaler01]).

Technical information systems are mainly used along the product generation process (PGP) for CAD, CAP, CAM, CAQ (cf. Fig. 1.4).

The splitting into the two core processes "customer order processing" and "product generation" and the related separation into commercial and technical information systems results in a variety of system interfaces. Redundant, inconsistent or outdated data as well as poor data quality (cf. [Lorenz00]) are the consequence. Often, they lead to instable processes[4].

Fig. 1.4 shows the interdependencies between the two core processes and the commercial and technical information systems considering the basic data.

---

1 See requirement 3.2.1; 3.2.2.

2 Data which are relevant on their own without relation to other data.

3 Describe the relations between the variants of master data.

4 See requirement 3.1.4; 3.3.3.

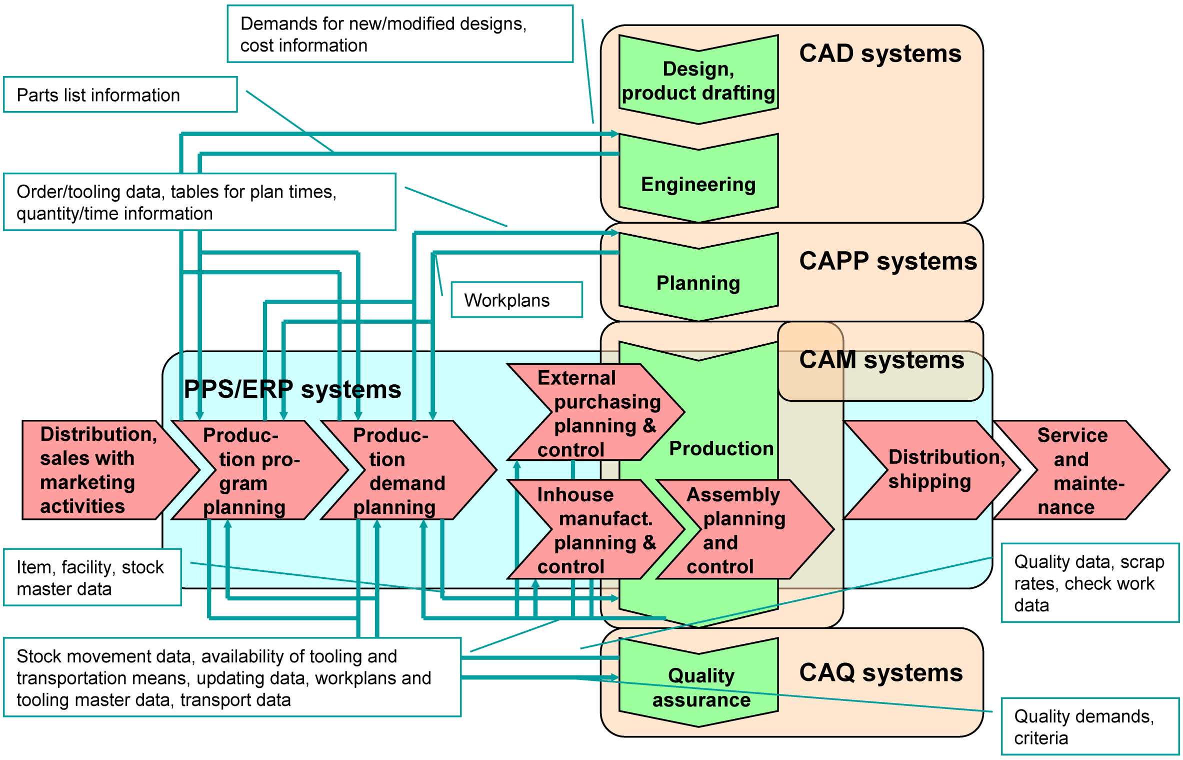


Fig. 1.4: Interdependencies between the core processes
of customer order processing and product engineering (Source [Daimle00])

For a few years, these technical information systems with their product and production describing and production controlling data have been integrated to an increasing extent with the introduction of PDM systems (cf. [Schött99])[1]. It is the objective to digitally map the products with the information relevant for their entire lifecycle (cf. [Paul86]). The central element has always been the product structure.

While first PDM solutions were close to engineering and often were restricted to a mere administration of drawings or CAD integration (cf. [Wijnke03]), today's PDM systems shall link internal and external processes with each other (cf. [Rybak02]).

In this context, PDM is defined as the management of product defining data (product model) in connection with mapping and managing technical/organizational business processes (process model). Product and process management together shall provide for a continuous reconfiguration of any design and manufacturing versions and levels across the entire product lifecycle (configuration model) (cf. [Eigner01]). Thus, product and process modeling are in a direct connection.

However, the originally anticipated targets and increases in efficiency have not yet arrived despite the introduction of PDM systems according to [Schött99]. The main cause is the fact that current PDM systems were developed based on conventional product structuring methods. These methods do not live up to the requirements of increasing variant diversity (cf. section 1.1.1). They allow for modifications of the work methods to a limited extent only.

1 See requirement 3.3.3.

Likewise, the most recent approach propagated under the keyword of PLM[1] which promises an expansion of the traditional PDM term towards a flexible, distributed application environment will not be capable of exploiting the existing potentials (cf. [Eigner01]).

The same applies for the linking of the technical and commercial information systems. According to [Rybak02], the integration of the ERP and PDM systems[2] has developed into a strategic aim for the automotive industry and is seen as a prerequisite for the complete realization of the CIM strategy (cf. [Schött99]). Nevertheless, the anticipated increase in efficiency has not yet arrived.

One reason is that existing systems or solution approaches start from conventional product structuring methods. The hierarchically designed product structures describe a product as the sum of parts or assemblies. The methodological approach does not consider the process information[3] (cf. section 2.5).

1.1.4 Summary

Automotive industry is subject to a permanently growing variant diversity and increasing product and process complexity at the same time.

Existing methods of product and process modeling have reached their limits long before. Based on hierarchically oriented product structures, products are still being described as sum of single parts and assemblies. Structural dependencies between the parts and major components cannot be imaged at all or in parts only. Likewise, an integrated documentation of process information is possible to a limited extent only. Existing ERP and PDM system solutions and the implemented methods start from these conventional methods. Therefore, they cannot live up to the requirements of variant diversity and designing flexible and efficient processes. Fig. 1.5 shows this interaction of variant diversity and process design with the information systems and the underlying methods of product and process documentation.

The potential inherent in the documentation as instrument of corporate management with effects across processes is fairly missed[4]. A continuous information network within the entire manufacturing company cannot be realized adequately. Despite the central meaning of the

1 Product Lifecycle Management is a useful term pointing to the fact that the thinking in areas and individual systems has to be replaced by a thinking in processes (cf. [SapPlm01]).
Others terms used in literature and by offerers are:
cPDM (collaborative Product Data Management by CIMdata)
VPDM (Virtual Product Definition Management by Gartner Group)
e-PLM (electronic Product Lifecycle Management by AMR) and
PDC (Product Definition and Commerce)

2 [Eigner01], e.g., described alternative integration approaches.

3 See requirement 3.1.1; 3.1.2.

4 See requirement 3.2.1.

product structure, product documentation is often treated as a passive and sequential writing down of the information collected in the manufacturing process.

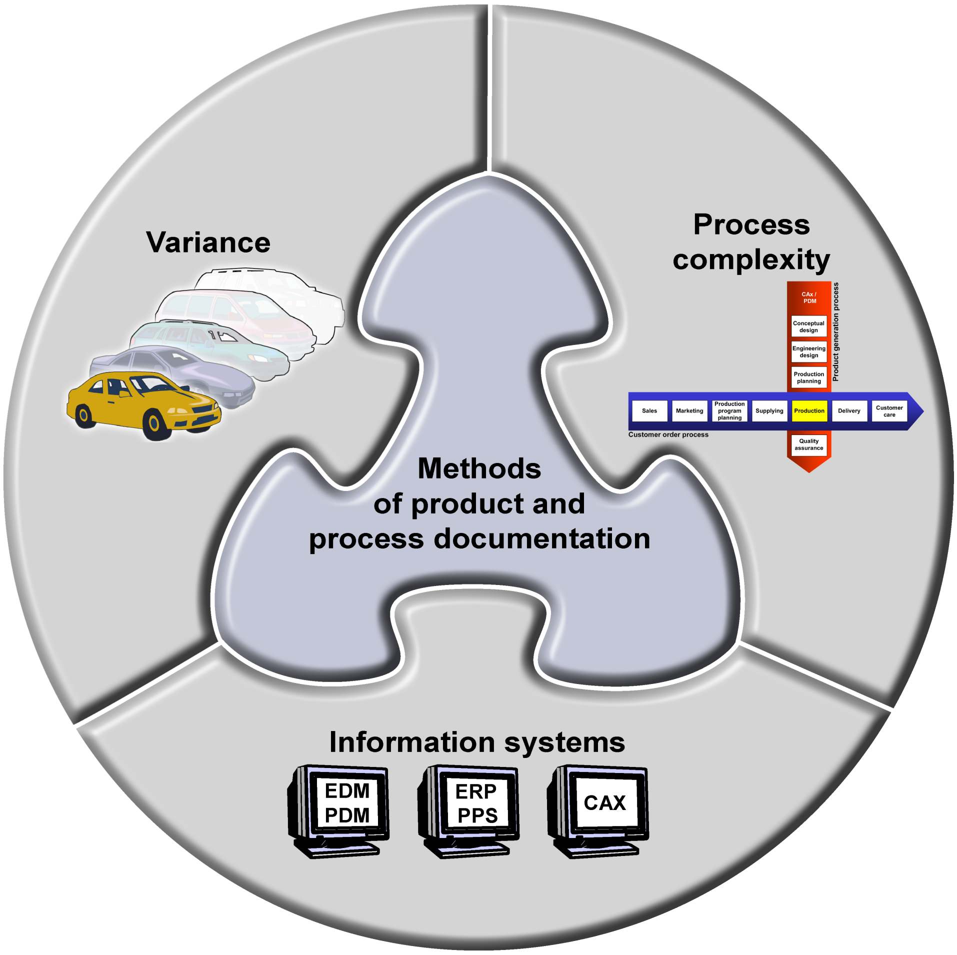


Fig. 1.5: Mutual effects of documentation
and process design, variant diversity and information systems

A future oriented documentation method must not be limited to a static writing down of the manufacturing process, but must work as an instrument to support a dynamic product and process management. It is a controlling instrument like this which can be used effectively as linking element between the numerous digital models and the (single) physical vehicle[1]. The prerequisite for such a management instrument is an appropriate method for an integrated product and process documentation.

1.2 Hypothesis

Increasing customer orientation and the resulting problems of variant diversity and product complexity correlates to the complexity of business processes in manufacturing companies. A parameter for sustaining national and international competitiveness on a long-term scale is the capability and fastness with which a company can react towards changes regarding products

1 See requirement 3.2.1.

as well as processes. The fastness of corporate reactions towards occurring changes to obtain a stable process situation again, depends on the product or process transparency. This transparency again depends on the method of product and process documentation.

1.3 Target and content of the study

The primary target of this study is to develop a consistent and continuous method for an integrated product and process documentation.

The term method means a systematic procedure or technique, employed by a particular discipline (cf. [WWWm1]). In the context of *Technical Documentation*, this method contains the complete description of product, process and resources information for all products of a company. Amongst others, the relevant data include here CAD drawings, parts lists, assembling plans and supplier data. The task of product and process documentation is to completely document all related modifications during the entire product life cycle. It accompanies a product during the different phases of the product generation process, i.e. from the decision to manufacture a product to the final specifications and project descriptions, from testing to series entry and the product discontinuation. As foundation of the entire product lifecycle in an industrial company, the documentation method must be capable of describing the information which are generated in the customer order process. With the understanding of an integrated (product generation and customer order) process, the product and process documentation must be able to map this information in a logical data structure.

Due to the critical business requirements[1] and the "historically grown" framework conditions of the product and process documentation used today, it seems only feasible to advance these existing concepts and methods. Another target of this work therefore is to consider and develop a migration strategy which lives up to real life demands.

The method of an interconnections documentation developed in this work provides the core of an integrated product and process data model (IPDM) as web of parts and interconnections. The study focuses on describing the methodological procedure to build and maintain this product web. Using a few application functions, like the change management, as example, the study shows how this integrated documentation is used properly.

In the sense of an enhancement of the documentation methods used today, the migration strategy required for the described interconnections documentation is provided. The initial (non-reducible) migration step is described.

The integrated product and process data model is verified in a prototype implementation. The methodological procedure of documentation is presented in the form of a business scenario.

1 For the Mercedes-Benz car manufacturer, e.g., up to a million customer orders have to be dissolved in two hours when determining the parts demand.

1.4 Structure of the study

This study on the interconnection based product and process documentation is divided into four main parts and seven chapters:

- Introduction and analysis of current methods of product and process documentation (chapters 1 and 2)
- Requirements on an integrated product and process documentation (chapter 3)
- Concept of the interconnection based product and process documentation including the migration strategy (chapter 4)
- Design and prototype implementation (chapters 5 to 7)

Following the representation in Fig. 1.6, the structure will be explained.

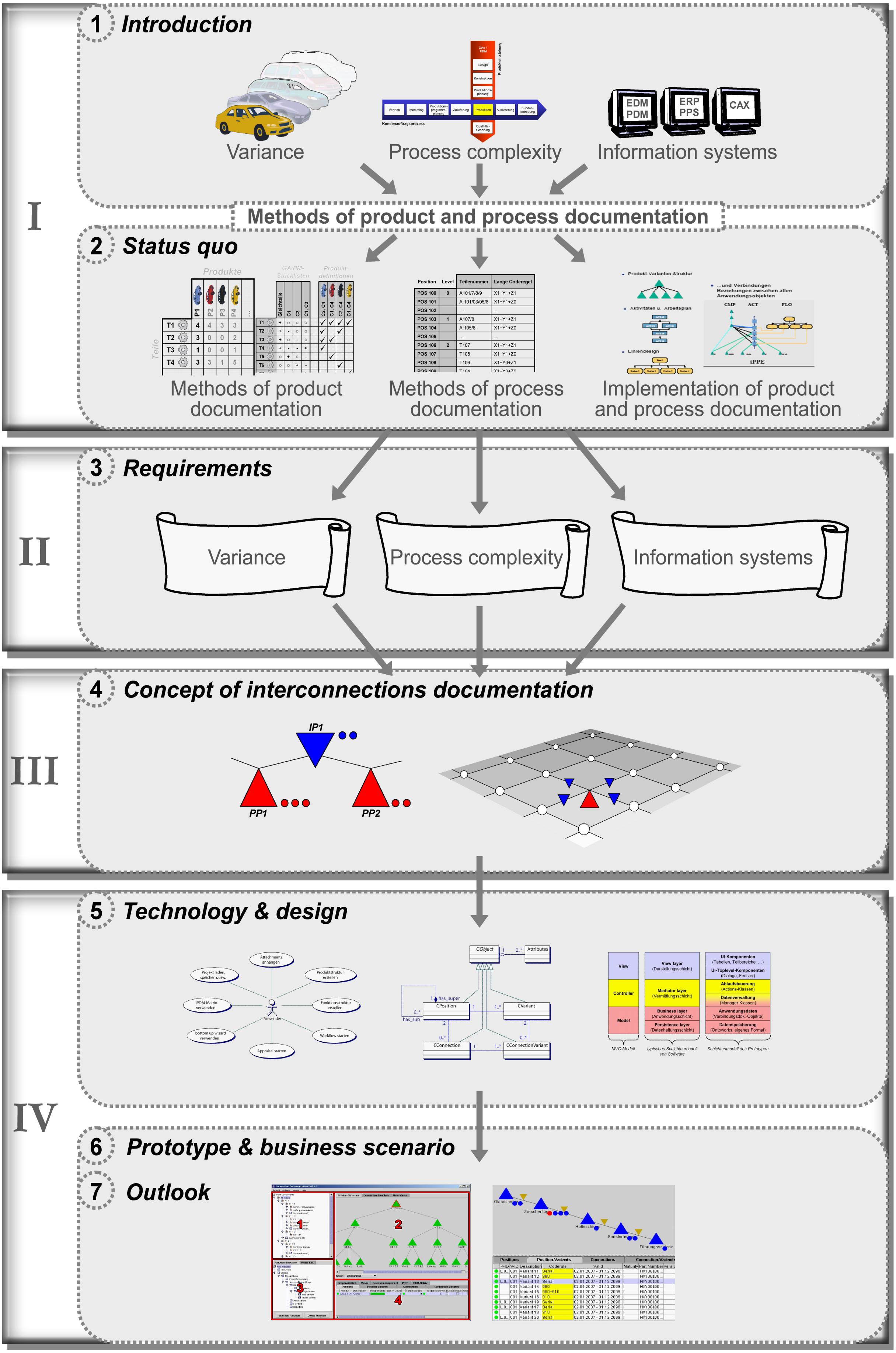


Fig. 1.6: Structure of study

The first main part (chapter 1) of the study deals with the current situation of producing companies in general and the automotive industry in particular. The problem of variance and complex processes prevailing in the companies in relation with the information systems will be analyzed to reveal the special significance the product and process documentation methods have in this context. Considering this knowledge, the documentation methods will be analyzed in the second chapter. At first, the methods of product documentation will be dealt with, followed by the methods of process documentation. The analysis of existing and implemented systems of integrated product and process documentation will consider solutions from corporate practice as well as scientific prototype implementations.

Based on the results of the first two chapters, the second main part of the study formulates the requirements placed on an interconnection based product and process documentation as seen from the problems of variance and process complexity.

The concept of the interconnections documentation is described in the third main part (chapter 4) of the dissertation starting with a layout of the methodological approach of this documentation form in section 4.1. Subsequently, the underlying concept of this documentation method will be detailed further. Sections 4.2 to 4.8 concentrate on describing the method, while section 4.9 deals with various applications (integrated change management amongst others). The third part closes with a description of a step-by-step migration concept in section 4.10.

The fourth and last part of this study addresses, amongst other topics, the design and basics for the prototype implementation of the documentation method (chapter 5). Focus will be on an object oriented design of interconnections documentation based on Unified Modeling Language (section 5.1). An appropriate IT architecture concept and the related technological basics for a prototype realization are developed in section 5.2 and 5.3. Chapter 6 presents the prototype implementation of the interconnections documentation. Section 6.1 describes the substantial application features followed by section 6.2 to introduce the business scenario "ITego". Based on a model vehicle, the product engineering process will be analyzed step by step in view of the interconnections documentation to evaluate the prototype implementation. The study on interconnection based product and process documentation closes with a summary and recommendations in chapter 7.

1.5 Fonts and formatting

Fonts and formatting styles are used to identify special texts as follows:

- Definitions are indented on either side.
- Quotes are embedded in double quotation marks followed by the corresponding source information.

- Literature references are embedded in square brackets. Please refer to the bibliography for the assignment of the reference ID to the exact title.
- Technical terms are written in italics.
- Contents of dialog boxes and fields are written in bold.
- For explanations of technical terms which are not detailed in the text, the reader is referred to the glossary (pp. 169).

2 Analysis of current methods of product and process documentation

This chapter addresses the analysis of the current methods of product and process documentation. At first, section 2.1 introduces the basics of product and process documentation. Sections 2.2 and 2.3 respectively will describe the basic concepts of product level and technical level of product documentation. To develop an integrated product and process documentation, section 2.4 analyzes the different methods of product documentation. Subsequently, the methodological concepts of process documentation are explained in section 2.5. The chapter will be closed by presenting examples of implementations of integrated product and process documentation in section 2.6.

2.1 Basics of product documentation

While section 2.1.1 at first defines product documentation and product and process documentation respectively, section 2.1.2 describes the tasks and targets of the *Technical Documentation*. Section 2.1.3 introduces the various levels of product documentation. As a supplement, the related manufacturing levels will be presented in section 2.1.4.

2.1.1 Method of product documentation

The term method originates from Greek (methodos) and means a systematic procedure, technique, employed by a particular discipline (cf. [WWWm1]).

In this context, the product documentation method means the type of procedure to completely describe the product information for the entity of all products of a manufacturing company.

Accordingly, the method of product and process documentation means the orderly and consequential procedure for a complete modeling of the product and process information for the entity of all products of a manufacturing company.

2.1.2 Tasks and targets of Technical Documentation

Technical Documentation comprises the complete description of product and process information. For [Ohl00], *Technical Documentation* is one of the fundamental and indispensable elements of any industrial enterprise. According to [Schött99], it represents a corporate success factor. Hence, *Technical Documentation* forms the foundation for the entire product lifecycle (cf. [Seybol97]).

The task of the documentation is to completely document all products of a company and all related modifications during the entire product lifecycle (cf. [VDIBer_97], [Ewers94], [Merced94]). Product documentation accompanies a product during the different phases of the product generation process, i.e. from the decision to manufacture a product to the final specifications and project descriptions, from testing to serial production and product discontinuation (cf. [Adam98]). Information obtained during the product engineering process forms the basis for all processes subsidiary to product engineering. Examples of relevant data are CAD drawings, parts lists, assembly plans, supplier data, etc. of all products developed, manufactured and sold by a company (cf. [Ewers94], [Merced94]).

2.1.3 Levels of product documentation

[Herlyn90] suggests to structure product documentation in three levels (cf. Fig. 2.1).

On the **product level**, communication between customer and manufacturer takes place. The product is defined, and all variants are specified (cf. [Herlyn90]). The product level, however, does not give any information about the composition and structure of the products (cf. [Ohl00]). The terms and concepts of the product level which are essential for the understanding of this study are explained in section 2.2.

The **technical level** documents the parts list information of the product description. This level renders the structural relations of the products where structural information is represented by the product structure or the methods of product documentation. The product structure links the technical level of product documentation both to the product level and the geometrical level (cf. [Ohl00]). Apart from this vertical integration, the product structure and the associated documentation method also provide the basis for a horizontal integration of the process information (cf. Fig. 2.1).

At this point, the method of an interconnection based product and process documentation which is described in this study comes into place. The different product structuring and documentation methods will be analyzed in sections 2.3 to 2.5.

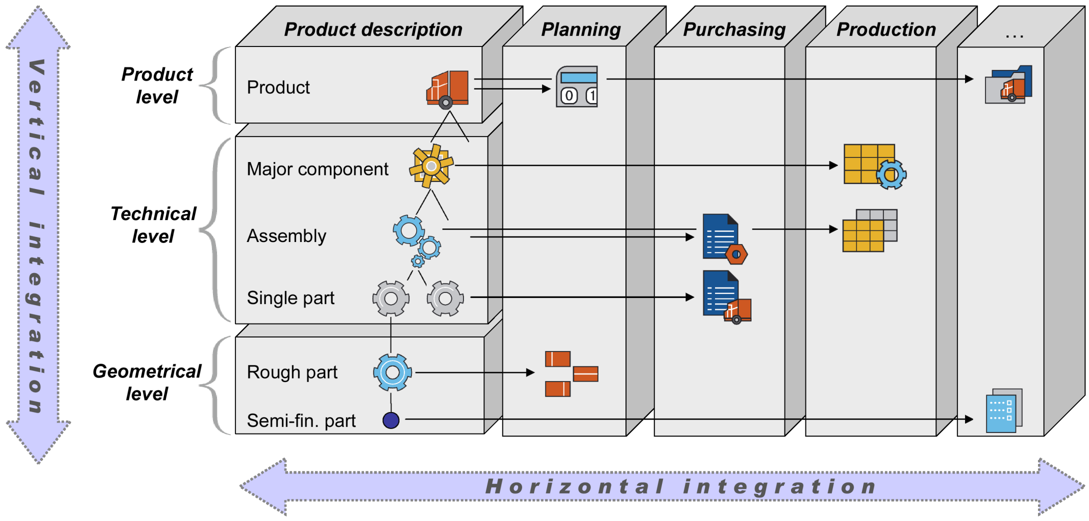


Fig. 2.1: The three levels of product description based on [Herlyn90]

The **geometrical level** describes the states of the parts until they have reached their ideal design status. Amongst others, this status is defined using the feature technology in the form of a CAD model or drawing (cf. [Ohl00]). Typical information is not limited to the identification of a part, but also include geometric information, i.e. shape, dimensions and tolerancing as well as physical properties of a part. The analysis of the geometrical level of product documentation will be set aside in this section. Chapter 4, however, will address an integration of the technical and geometrical level when introducing the concept of the interconnections documentation (cf. section 4.3).

2.1.4 Manufacturing levels of the product description

During the manufacturing process, a product passes different manufacturing levels (cf. [VDIT77_76], [VDIT28_72], [DIN_77]). In automotive industry, the following levels are differentiated:

- product
- major component
- assembly
- (single) part
- rough part and semifinished part

For a detailed description of the different manufacturing levels and their differentiation, the reader is referred to the glossary and advancing literature (cf. et al. [Lingna94], [DIN_77], [VDIT77_76], sim. [Ungehe86], [Heinis91], definitions in [VDIT77_76]).

2.2 Basics of the product level

This section describes the essential terms (see section 2.2.1) and concepts which are required to manage variant diversity of vehicles and communicate it to the customers. Sections 2.2.2 to 2.2.7 will address the special meaning of the options to configure a vehicle and their role as linking element between product level and technical level (see section 2.2.8).

2.2.1 Structuring terms on the product level

To be able to communicate the tremendous number of vehicle variants of a car manufacturer to customers, a systematic reviewing of the products is carried out on the product level regarding a clear hierarchical structuring. The terms which have been established, like "model range", "model subrange", "type", "model" and "variant", are described in detail in the glossary. Fig. 2.2 shows their hierarchical ranking.

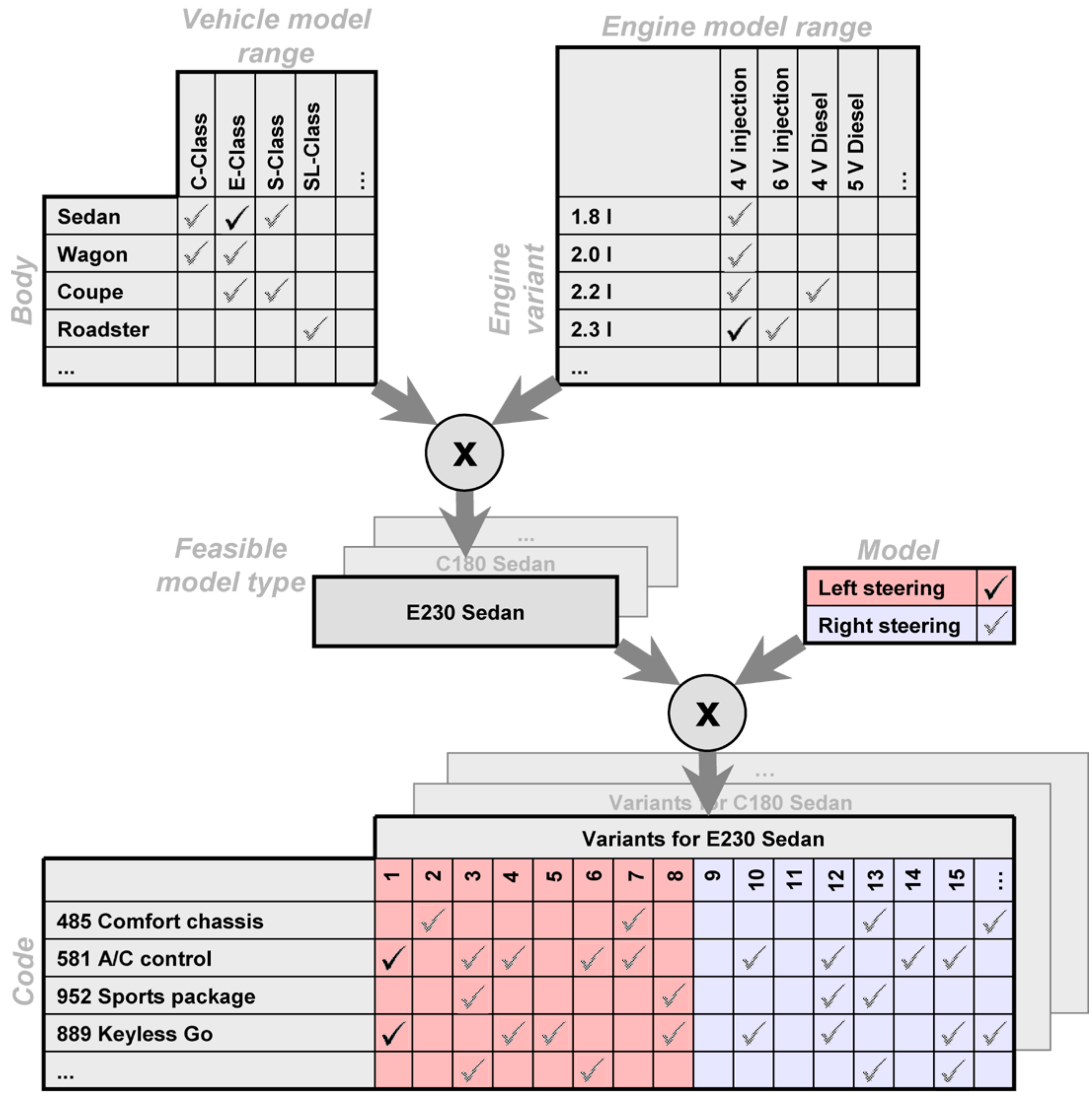


Fig. 2.2: Breakdown of hierarchical product structure using Mercedes-Benz as example

2.2.2 Vehicle configurations based on options

Customers who want to buy a vehicle select the desired vehicle from a variety of variants offered to them based on the ideas the customers specify as their preferred vehicle characteristics. From the customer's point of view, characteristics are described by options[1] which will be translated into codes (cf. [Ohl00], [Audi96], [BMW96], [Volksw96], [Merced96a]). All options or codes offered are documented in the form of an option or code list in the sales documents (cf. [Ohl00]). Based on the sales documents[2] or via IT supported product configurators, the customized vehicle is specified. This specification can be realized together with a sales representative or via the internet, for example.

Today, vehicle configurations based on options/codes are considered to be standard in automotive industry. There are only minor deviations between the different manufacturers. A well performing options management between the functions of sales, engineering and operations represents a special challenge. It is all the more challenging for manufacturers with rich variance, as is the case at Mercedes-Benz for example.

2.2.3 The technical content of options

According to [Ohl00], options describe the properties of a product from the customer view. Options do not contain information about the actual technical content of the product in the form of single parts or assemblies. Options therefore do not describe one or several parts, but functions. It is only with the assignment of an option or its code to a defined model range, or sometimes even with the assignment of the code to a specific model that the demanded parts are determined unambiguously (cf. [Landvo94]).

Selecting the options or codes does not only express the customized vehicle but also codes the vehicle at the same time (cf. [Maeker97], [Schöns85], [Zieman84]).

2.2.4 Mandatory and optional options

According to [Ohl00], it is favorable to classify codes by dividing them into options from which the customer "can" select one or several options and those options where the customer "must" decide on exactly one feature.

Apart from differentiating by mandatory and optional options, [Herlyn90] suggests to bundle options which exclude themselves in option classes. For each vehicle configuration, one option may be selected as maximum for each option class. Standard option means that option of an option class which is assigned to a product by default.

1 In the following, the terms option, feature or option are used as synonyms.

2 Sales pocket books or price lists, e.g.

2.2.5 Basic and additional options

For products with many variants, the model range or model and the corresponding product variants are distinguished. According to [Herlyn90], this structure also serves as basis for a classification into basic and additional options. Basic options determine the model range, while the additional options define the variants of the model range already specified. Hence, the basic options represent the unequivocal reference point for all additional options. [Ohl00] compares the basic options to a frame all product variants of a model range move within. The additional options merely reflect the layout inside this frame.

To define a product variant, a basic option is always mandatory, but not additional options. The basic option represents a mandatory option, and additional options correspond to optional options (cf. [Herlyn90]).

2.2.6 Standard and optional equipment

The vehicle version with its features resulting from the specification of all mandatory options and choosing one option variant each from the options available at no extra charge without ordering additional optional options is referred to as standard equipment. Additional features resulting from ordering optional options or choosing option variants at extra charge for the mandatory options are referred to as optional equipment (cf. [Fische93]).

2.2.7 Implicit and explicit product documentation

In principle, there are two ways to define a product using options.

The **implicit product documentation** starts from a basic product which contains certain characteristics as default right from the start. These characteristics are contained in the basic product implicitly and do not have to be mentioned separately. In this case, the product definition has to be supplemented by specifying the respective additional options only when a version different from the basic one is desired. As a consequence, implicit product definitions must contain one option as minimum, the basic option. They are expanded as required by specifying other additional options. Implicit product definitions are of a variable length where the maximum length of the product definition is limited by the number of option classes (cf. [Herlyn90]).

An **explicit product definition**, in contrast, does not start from a basic product. There is also a basic option, but it does not imply standard options. The basic option defines a single property only and not a complete product as it is possible with implicit product definitions. Hence, the standard options have to be specified explicitly, even if they merely mean the "nonexistence" of a property (cf. [Herlyn90]). With an explicit product definition, it is necessary to choose exactly one option from each option class. As a consequence, the length of the product definition is constant being defined by the number of option classes.

2.2.8 From customer configuration to a production order

A vehicle specified by a customer does not represent a fully qualified data record for ordering. To generate an order which can be processed in the manufacturing plant and determine the vehicle specific parts list, the data record has to be checked for its feasibility and supplemented by additional information. In this context, the product overview is a central component of the product level. Amongst others, it manages the feasibility rules providing the database for all vehicle configurations. The documentation of the feasibility rules[1] in the product overview and the resulting "decoupling" from the product structure cause an interface (which is not necessary) representing a potential source of errors[2]. Deficiencies in documenting feasibility may lead to errors in the BOM explosion resulting in a faulty calculation of the parts demand.

Due to an insufficient integration of parts list information into the processing data of production planning, amongst others, feasibility restrictions on the production side are not available to the required extent for a communication between manufacturer and customer[3].

2.3 Basics of the technical level

This part of the work introduces the key elements of the technical level of product documentation. At first, section 2.3.1 defines the terms of master and structural data. In literature, the terms parts list, product structure and product schema are often used as synonyms, but in sections 2.3.2 to 2.3.4 they will be distinguished properly against each other. Section 2.3.5 deals with the graphical representation of structural information.

2.3.1 Master and structural data

The basic data to be managed in the context of PDM and ERP systems can be divided into master and structural data (cf. [Grupp95]):

- Master data mean those data which are relevant on their own without being in relation to other data. Amongst others, the item master data belong to the key master data.
- Structural data are used to create relations between the values of master data. For [Eigner01], [Bender04], the different forms of a parts list belong to the key types of structural data apart from the work planning data.

While the terms parts list, product structure and product schema are widely used as synonyms, they will be described in detail in the following to distinguish them properly.

1 Amongst others, the objection free determination of option classes or the assignment of the options to the relevant model ranges, models, product variants.

2 See requirement 3.1.4.

3 See requirement 3.1.1; 3.1.2; 3.2.1; 3.2.2; 3.2.7.

2.3.2 Parts list

According to [DIN_77], a parts list represents a formally structured directory for an object which is complete for the respective purpose. It contains all belonging objects specifying nomenclature, item number, quantity and unit (cf. [Zäpfel91], [Ewers94], [Brändl95], [Eigner01]).

Parts lists thus create a link between the customer order and the required parts (cf. [Johnso96], [Ottens96]). They are used in different forms whenever end products are composed of several initial parts (cf. [Schött99], sim. [Specht94]).

For [Ohl00], a parts list serves as the necessary prerequisite to provide the proper single parts, assemblies and components in the proper quantity at the proper time at the proper place in the proper quality at the proper costs.

According to [Herlyn90], an individually customer specified vehicle consists of several thousands of parts. In this context, [Hirzel95] states the number of parts per order in a range of 15,000 for the passenger car division at Mercedes-Benz. The exchange of a single part only or a small group of parts results in a new vehicle variant or a different product.

Frequently, parts lists also describe structural information of a product. While parts list forms, such as quantity lists, describe information regarding the quantitative composition of products, parts list forms, like the structural parts list, contain structural information to render the structure of a product.

2.3.3 Product structure and product schema

According to DIN 6789, each product or its elements are based on a structure which is called hierarchical or product structure. A general distinction can be made between single level and multilevel products (cf. [Warnec93], sim. [Eigner01] and [Brändl95]). Product structures describe the assignment of product components to each other (cf. [Herlyn90]). This includes structural relations of parts and assemblies to the product[1] as well as structural relations of parts and assemblies of a superior or lower level to each other. Thus, product structures define the structure of assemblies from the lowest level up to the product level. (Anytime), they reflect the current relations to the part master records.

The levels thus linked to each other are part of the documentation. They are realized as structural relations of the type "belongs to" (component use) or "consists of" (component explosion).

This type of product structuring only allows to understand and document a vehicle as the sum of its structural components (major components, assemblies, (single) parts, raw material or semifinished part). A structural associativity between different components of the same level

1 The information about the superior assemblies in which a specific part or component is used is called part usage.

is not possible or only via a superior level. Section 2.4.6 addresses the general problems inherent in this type of relations in the context of an integrated product and process documentation[1].

Within a company, product structures are of central significance. According to [Garwoo97], product structures have existed as long as manufacturing companies do. As a central means of product documentation (cf. [Ohl00]), the product structure is the backbone of any manufacturing company (cf. [Brändl95]).

[Frohma88] considers the product structure to be responsible for a complete mapping of all products, their structuring, and components. This wider definition not being limited to a product group (model range e.g.) addresses the function of a product schema. A product schema contains the entire product portfolio of a manufacturing company. Product schemas can be seen as over-determined product structures. They can also be understood as classification schema. In the case of automotive manufacturers, they include vehicle functions which mutually exclude each other for the product structure of an individual vehicle model range or cannot be assembled together.

The extent to which the issue of a suitable product structuring evokes controversy and how close the ties are between the terms BOM/parts list and product structure/product schema is reflected in the following quote by [Garwoo97]: "*Beginning a discussion on how to structure the bill of material is a good way to start a fight in a bar ...*"

2.3.4 Product schema versus product structure and parts lists

In the sense of PDM, [Schött99] considers the product structure to be the leading business object compared to the parts list. The product structure can be understood as information carrier for the parts list.

The parts list is generated by passing through the product structure "from the top to the bottom", i.e. from the product downward to the single parts and their initial materials (cf. [Eigner01]). It is hence possible to derive or generate the parts list for one or several assemblies or even an entire product from the product structure. The product schema plays a special role in this context containing the entity of all vehicle features which can be called a master structure. The product structures are derived from this schema thus supporting the commonality idea[2].

The relations between product schema, product structure and BOM are shown in an overview in Fig. 2.3.

1 See requirement 3.1.2.

2 See requirement 3.1.5; 3.1.7; 3.1.8.

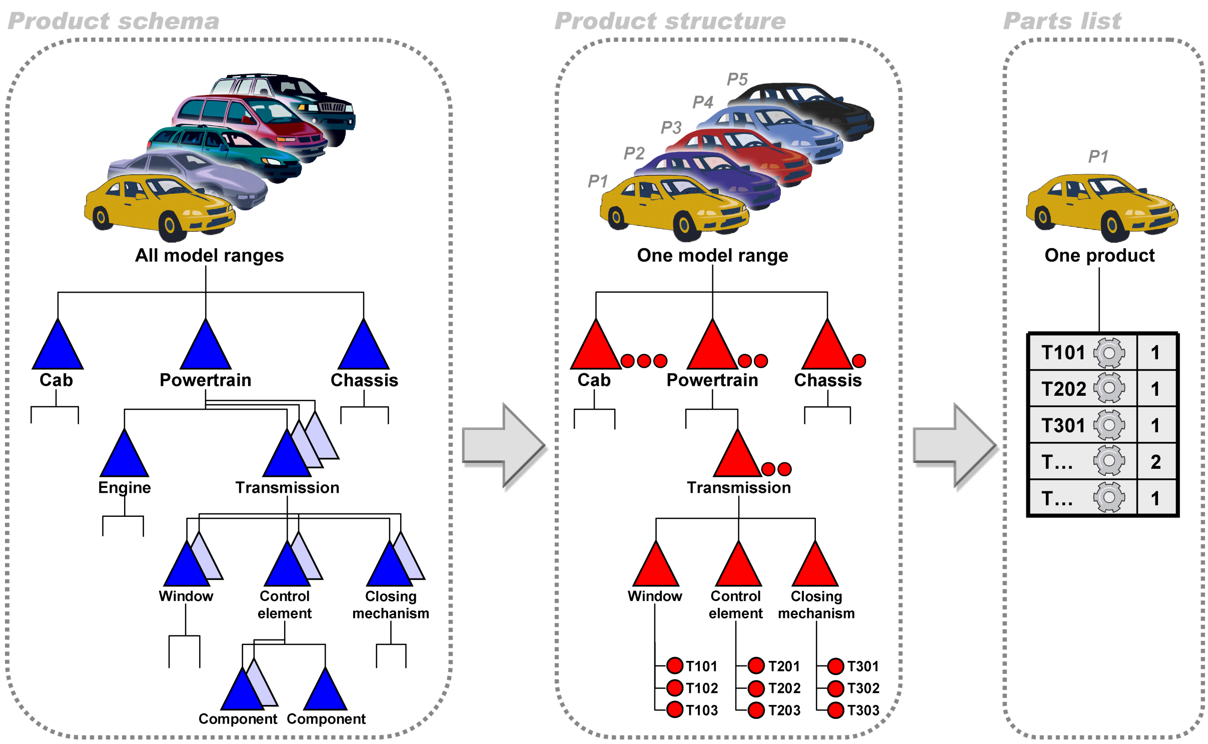


Fig. 2.3: Product schema versus product structure and parts list

For the preparation of manufacturing orders, the calculation and planning of the parts to be procured, it is indispensable to know the composition of the end products and provide the data as IT information. The generation of the parts lists from the product structure is the task of the parts list processor.

For this study, the term parts list shall be understood in accordance with [DIN_77]. Parts lists are generated based on a concrete vehicle configuration where the items are derived from the product structure. The product schema contains the entity of the features of all product structures in the sense of a master structure. Accordingly, the product structure manages all theoretically possible product variants of a product.

2.3.5 Types of graphical representations of product structures

Due to missing graphical tools, structural data were often represented as lists in the past. With the development of graphical user interfaces and the deployment of browsers, list representations have increasingly been set aside to the benefit of graphical forms of product and parts list structures[1]. For [Eigner01], the Gozinto graph (the part that goes into it) is considered to be the archetype of a graphical product structure representation.

1 See requirement 3.3.1; 3.3.3.

2.4 Methods of product documentation

Starting from the parts list classification in section 2.4.1, sections 2.4.1 to 2.4.7 will discuss the various parts list methods or product structures. Section 2.4.8 gives details regarding the selection of the structuring level. This part of the study is concluded with a critical reviewing of the current parts list methods in section 2.4.9.

2.4.1 Classification of the parts list methods

Fig. 2.4 shows an overview of the different parts list methods.

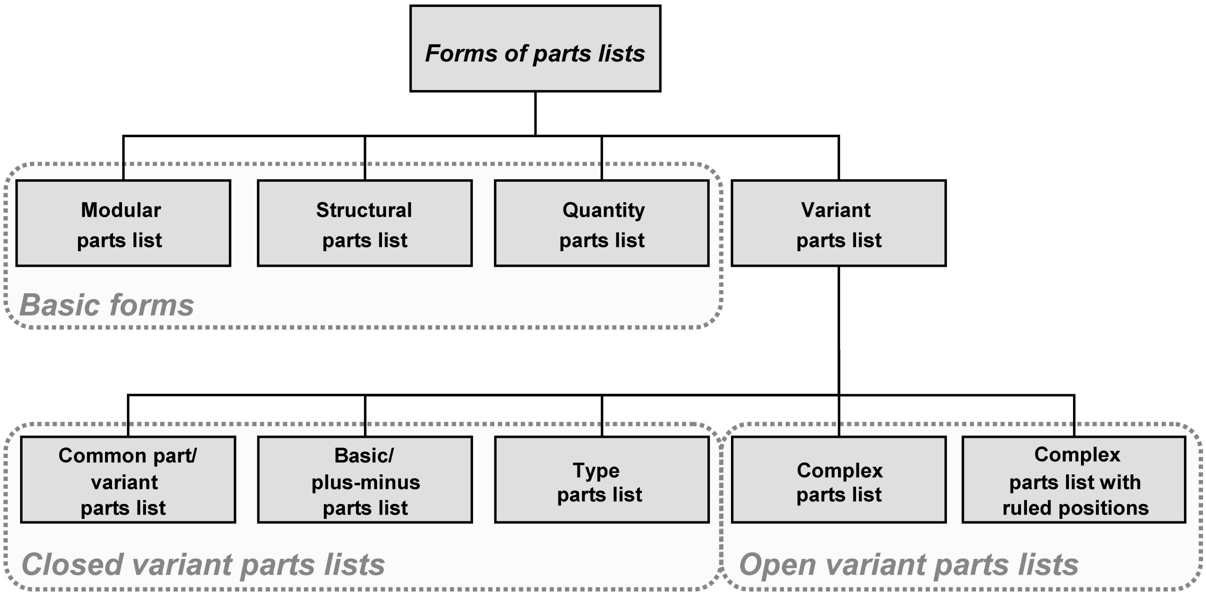


Fig. 2.4: Overview of parts list methods

The different parts list types or rather the underlying methodological concepts can be classified into basic forms of parts lists and variant parts lists. The variant parts lists, in turn, can be distinguished by closed and open variant forms. Apart from the forms described here, a variety of other parts list types exists. Frequently, these types are mixed forms of the approaches described here or represent theoretical advancements without any significance for practical worklife. To give an example, the WIBER method described by [HamiNo07] shall be mentioned here.

2.4.2 Basic forms of parts lists

The simple forms of basic parts lists include the

- quantity parts list
- modular parts list
- structural parts list

The **quantity or overview parts list** (cf. Fig. 2.5) is seen as the simplest form of a parts list (cf. [Grupp89]). A quantity parts list is an unstructured representation simply listing the quantities of the used elements without a specific order (cf. [Grupp89], [Zäpfel91]). Quantity parts lists simply represent a directory of the item numbers contained in a product without any information about the product structure (cf. [Zäpfel91]). For a product, each item number is listed only once with accumulated quantities, even if the item number occurs at several positions in the product structure as repeat part (cf. [Grupp89]).

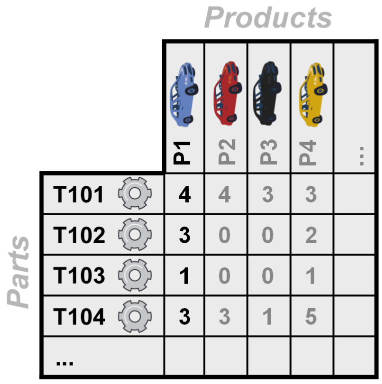


Fig. 2.5: Quantity parts list

The **modular parts list** (cf. Fig. 2.6) is a two-level parts list which always lists those components only which are directly required to manufacture a product. The components or parts of the hierarchically lower structural level are documented with their quantities in positions (cf. [Eigner01], sim. [Grupp95], [Pahl05], [Grupp89]). A multilevel product has to be represented via several modular parts lists (cf. [Roh98], [Zäpfel91]). The complete design and manufacturing context can only be concluded by adding all modular parts lists in the form of a structural parts list (cf. [Grupp89]).

Due to the very resource consuming (multilevel) parts lists explosions, computing times to calculate the parts demand are very long (cf. [Herlyn90]). Modular parts lists can be used to derive the structural and quantity parts lists as basic parts lists for IT applications (cf. [Eigner01], [Grupp89]).

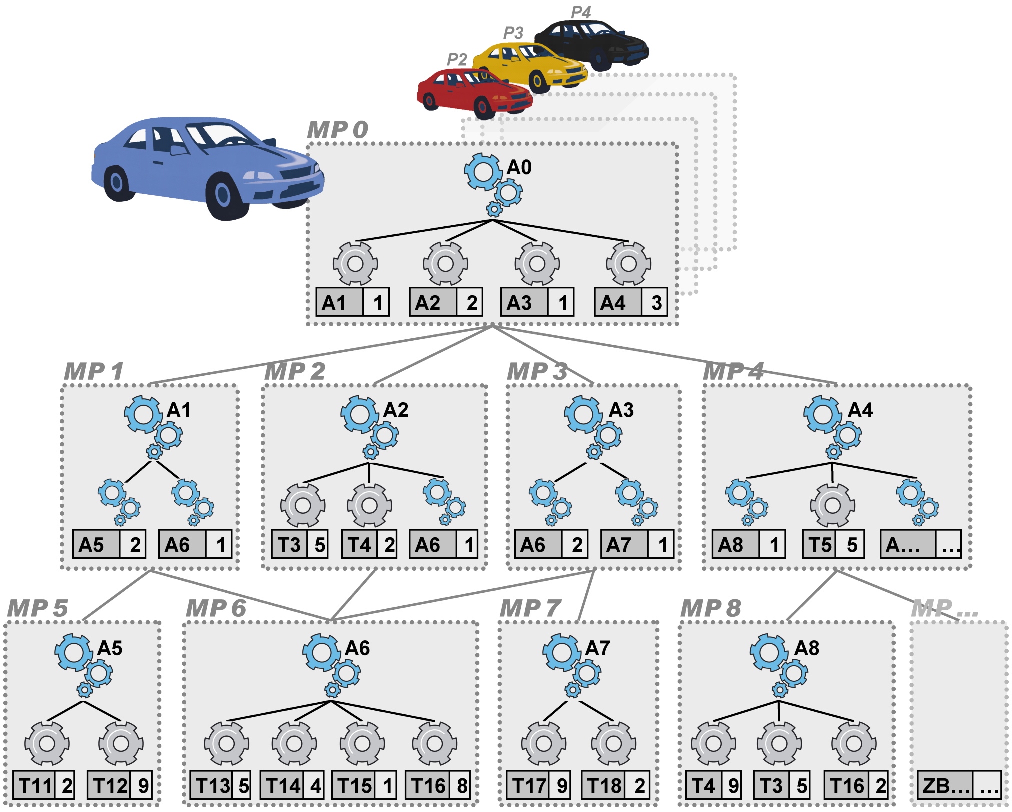


Fig. 2.6: Modular parts list

The **structural parts list** (cf. Fig. 2.7) allows to capture a product across all hierarchical levels in a single parts list (cf. [Eigner01], [Roh98], [Zäpfel91]). The structural parts list shows the entire manufacturing structure of a product with all used assemblies and parts in a continuous sequence. The structuring of the listed assemblies and parts corresponds to the planned manufacturing workflow (cf. [Grupp89]).

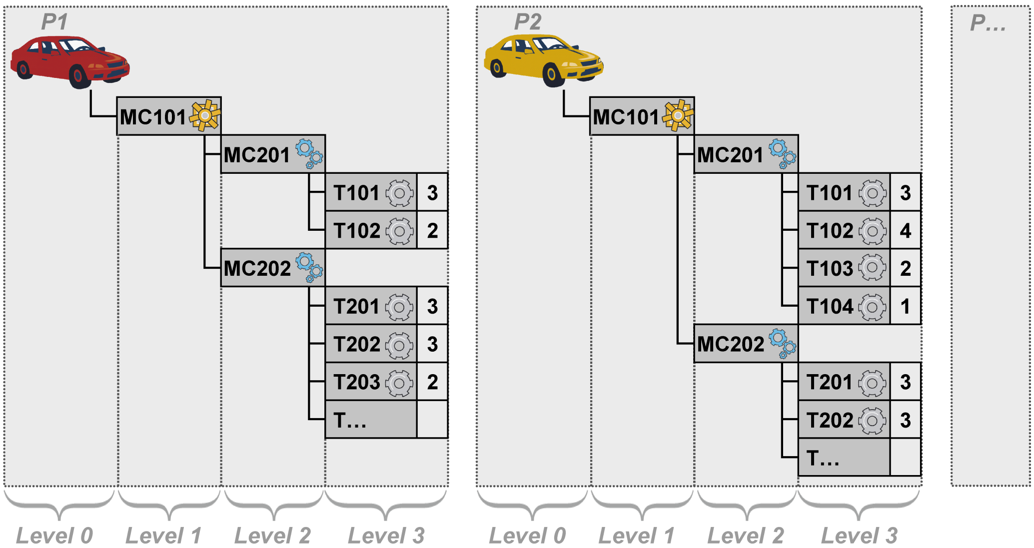


Fig. 2.7: Structural parts list

This parts list type unites the two other basic forms since both the quantities of all components of a product and the entire hierarchical structure of the components are visible (cf. [Eigner01]). For products with a large number of parts list positions, however, this type of a parts list becomes confusing (cf. [Grupp89]).

Final note

The number of the theoretically possible vehicle variants is very large compared to the number of variants occurring in practice and to be produced. In the run-up, however, the vehicle variants which the customers will order are not known. Hence, the parts lists must be built to allow for any theoretically possible variant (cf. [Herlyn90]).

While the individual vehicle variants often differ in only but a few parts or assemblies, data redundancy[1] between the individual parts lists is very high (cf. [Hübner75]). The documentation of all product variants in one or several parts lists is not sensible or even impossible according to [Brändl95][2] (cf. [Herlyn90], [Ploenz98], [Ohl00]).

2.4.3 Variant parts lists

Compared to the basic forms of parts lists, the variant parts list represents a special form of a parts list.

According to [DIN_77], the variant parts list is a collection of several parts lists on a form to be able to list different objects with a typically high share of identical components together in a list.

Variant parts lists result from the demand for an economically designed product structure which can map a large number of products while keeping the volume of different product structures as small as possible (cf. [Ohl00]).

The method of a variant parts list is always used sensibly when different products with a high share of identical parts or assemblies are to be mapped (cf. [Pahl05]). In such a case, it is more expedient to represent the variant parts in a shared product structure than completely document each product variant individually.

Customer wishes regarding the new product are described by coding the properties using options or codes.

The parts and assemblies installed into the corresponding product are derived based on the selected options exclusively.

Variant parts lists play an important role within product design (cf. [Eigner01]). A general distinction is made between variant parts lists of the

- closed type and
- open type

1 See requirement 3.1.4; 3.1.7.

2 See requirement 3.3.3.

2.4.4 Closed variant parts lists

The simple or closed variant parts lists include the

- common parts/variant parts list
- basic version/plus-minus parts list
- type parts list

The common parts/variant parts lists or basic version/plus-minus parts lists avoid redundant data by summarizing parts and assemblies which are used in several variants at the same time For each summary, a separate parts list is generated. This method offers the advantage that it is no longer necessary to list all parts and assemblies for any specific variant, but only the differing ones which are listed in the variant parts lists (cf. [Herlyn90]).

Product variants of the **common parts/variant parts list** (cf. Fig. 2.8) are represented as follows (cf. [Herlyn90]):

- Parts which are valid for all variants are listed in the common parts list.
- Parts which are valid for specific variants only are managed in the variant parts list.

With this method, the basic version parts list to which the variants refer does not consist of a complete product, but of the common parts from all variants of this model range (cf. [Grupp76], [Roh98], [Eigner01]). The common parts list is a more or less random collection of parts without a technical or functional relationship (cf. [Grupp76]).

Parts		Common parts/variant parts lists										Product definitions				
Position	Part no.	Common parts	C1	C2	C3	C4	C1, C2	C1, C4	C3, C2	C3, C4	...	P1 C1, C2	P2 C1, C4	P3 C3, C2	P4 C3, C4	...
S_POS 100	T102	+	o	o	o	o	o	o	o	o		✓	✓	✓	✓	
S_POS 101	T104	o	+	o	o	o	o	o	o	o		✓	✓			
S_POS 102	T103	o	o	+	o	o	o	o	o	o		✓		✓		
S_POS 103	T107	o	o	o	o	o	+	o	o	o		✓				
S_POS 104	T105	o	o	o	o	o	o	o	+	o				✓		
S_POS 105	T106	o	o	o	o	o	o	+	o	o			✓			
S_POS 106	T107	o	o	o	o	o	o	o	o	+					✓	
S_POS 107	T108	o	o	o	+	o	o	o	o	o				✓	✓	
S_POS 108	T110	o	o	o	o	+	o	o	o	o			✓		✓	
S_POS ...																

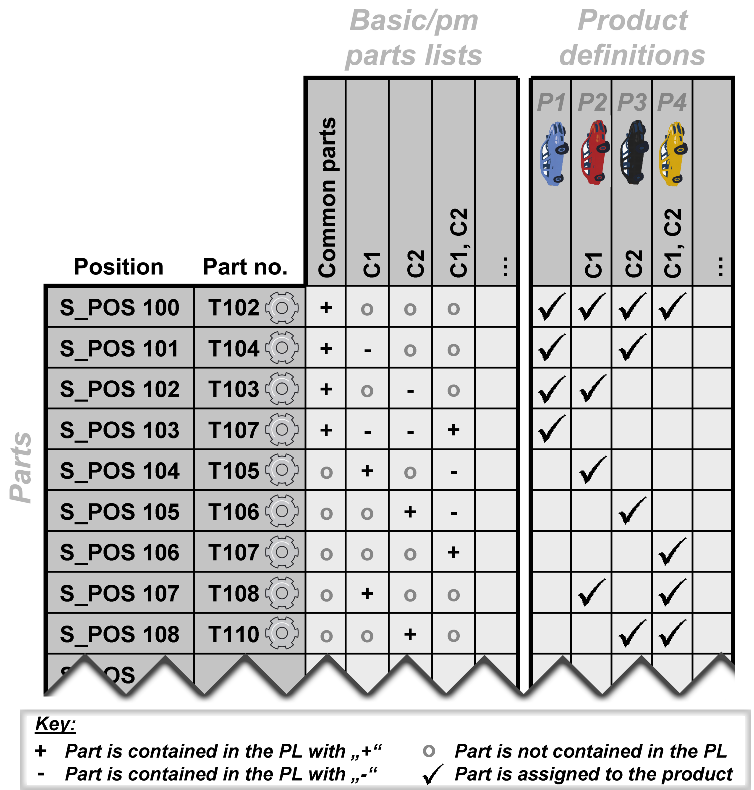
Key:
+ Part is contained in the PL with „+“
o Part is not contained in the PL
✓ Part is assigned to the product

Fig. 2.8: Example of common parts/variant parts list based on [Herlyn90]

Grouping the common parts to a basic or shrink variant very rarely represents an assembly in manufacturing terms[1].

The plus-minus method knows **basic version and plus-minus parts lists** (cf. Fig. 2.9). The basic version parts list documents the product structure of a vehicle in its basic or standard equipment. Accordingly, the plus-minus parts lists describe the part uses which are added (plus positions) or removed (minus positions) due to optional equipment in relation to the basic version parts list (cf. [Brändl95], [Herlyn90], [Roh98], [Grupp89]).

Plus-minus parts lists only map those parts which are added (plus) or removed (minus) for the variant. The description of a product variant, therefore, is distributed to the basic version and the respective plus-minus parts lists.



Position	Part no.	Common parts	C1	C2	C1, C2	...	P1	P2 C1	P3 C2	P4 C1, C2	...
S_POS 100	T102	+	o	o	o		✓	✓	✓	✓	
S_POS 101	T104	+	-	o	o		✓		✓		
S_POS 102	T103	+	o	-	o		✓	✓			
S_POS 103	T107	+	-	-	+		✓				
S_POS 104	T105	o	+	o	-			✓			
S_POS 105	T106	o	o	+	-				✓		
S_POS 106	T107	o	o	o	+					✓	
S_POS 107	T108	o	+	o	o			✓		✓	
S_POS 108	T110	o	o	+	o				✓	✓	

Fig. 2.9: Example of a representation of a basic version parts list and a plus-minus parts list (based on [Herlyn90])

The effort to generate this type of parts list is lower compared to that for common parts/variant parts lists, since the composition of a new variant does not touch the structure of the basic model or the existing variants. The individual plus-minus parts lists can be built independently from each other without touching the other plus-minus parts lists. For plus-minus parts lists for combinations of options, however, the plus-minus parts lists of the individual options must be available already.

1 See requirement 3.1.2; 3.2.3; 3.2.7.

Several similar parts lists can be grouped to a **multiple/type parts list** with several quantity fields. Homogeneous products with different single parts can be assigned to a joint model group. There are to general ways of representing variants when using several quantity columns:

- The first quantity column contains the quantities of the common parts. The subsequent quantity columns for the variants only contain the quantities deviating from the basic variant (type parts list) (cf. Fig. 2.10 a).
- Each quantity column shows the complete required number of pieces (multiple parts list) (cf. Fig. 2.10 b).

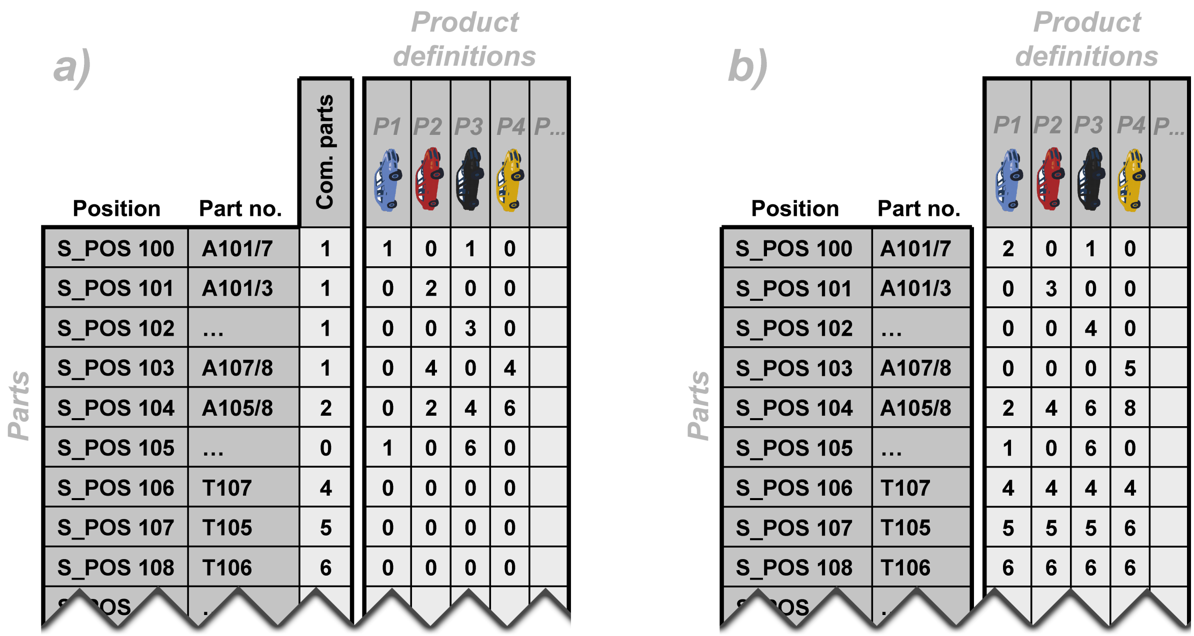


Fig. 2.10: Example of a multiple parts list
based on [Herlyn90]

Final note

Closed variant parts lists bear a few decisive disadvantages for their practical application in industry. The closed variant parts lists are little effective for variants with minor deviations only. If the variants, for example, differ exactly in one part or assembly only, groupings are not possible. In this case, the variant representation is not applicable (cf. [Herlyn90]).

Another disadvantage of the separate variant parts lists is that the contained parts and assemblies are grouped arbitrarily, i.e. they do not make sense from a technical point of view. Hence, they are without any relation to manufacturing or assembling[1].

To link the separate parts lists among each other, additional pseudo parts and fake positions are required. Likewise, these fake positions do not convey any technical information.

1 See requirement 3.1.1; 3.1.2; 3.2.3; 3.2.5; 3.2.7.

In practice, IT implementations based on the concept of closed variant parts lists are available to a very restricted extent only. Hence, closed variant parts lists are to be found very rarely in industry[1].

2.4.5 Open variant parts lists

With the mapping of products based on the method of the **open variant parts list**, all variants belonging to a product group are mapped together in one product structure. The parts list information is not distributed to a variety of different parts lists, but stored in a single structure.

This parts list does no longer contain products which are now documented on the product level, i.e. the product is removed from the technical level. In contrast to the "item number determined variants", the open form is "order determined". For a specific order, the required parts and assemblies for a concrete product variant are selected dynamically considering the customer order specifications.

The plain variant parts list is expanded by conditions. This information defines the conditions when a part or assembly is used for a product. The conditions are phrased using the options which create the relationship between the parts and assemblies and the product (cf. [Herlyn90]).

Hence, expanded variant parts lists are no longer generated and structured based on the analytical, but synthetic method.

Parts list types based on the method of the open variant parts lists are also known as selection or maximum parts lists (cf. [Ohl00]).

Open variant parts lists offer the substantial advantage that product variants can be managed in a very condense way. Besides, the integration of new, additional variants is easy. In contrast to the closed variant parts lists, the product variants to be documented are not completely predefined (cf. [Kurbel95]). For [Meinin94], such a parts list represents all "feasible" variants of a product type which were designed by engineering.

The best known examples of open variant parts lists and which are used most frequently in industry are

- the rule-based complex parts list
- the complex parts list with ruled positions based on it

The following sections will explain the methodological approaches of the rule-based complex parts lists in detail.

1 See requirement 3.3.3.

2.4.6 Method of rule-based complex parts list

For [Ohl00], **rule-based complex parts lists** are characterized by the fact that each position of the list stores information about the conditions on which the respective part or assembly will be used in the next higher structural level.

At a specific point in time, product variance is caused exclusively by the options or codes introduced in section 2.2.2. Therefore, these options represent the crucial components of the rules or conditions which affect the parts used. Rules form the core of this method. They are an atomic constituent of the logical condition which defines for each part the reason why it is used (cf. [Ohl00], [Brändl94]).

The conditions which must be met for a specific part to be used in a vehicle, are documented in the form of Boolean expressions, also referred to as code rules.

> Code rules consist of the codes (options), rule elements, and the Boolean operators listed in Table 2.1 which are linked in accordance with the principles of Boolean algebra (cf. [Ohl00], [Merced95b]).

Code rules can consist of no code (empty code rule ";"), a code (single code "C1"), or several codes linked together (code rule "C1+C2"). The code rules determine the condition when a part is to be used in a product variant. Code rules do not only assign parts and assemblies to an individual product variant, but to different product variants at the same time (cf. [Herlyn90]).

Table 2.1: Boolean parts list operators
(based on [Herlyn90])

Operator	Meaning
;	Always valid
+	Logical "and"
-	Logical "not"
/	"Exclusive or"

The Boolean expressions can be interpreted as rules that are used to control when which parts are used in the respective end product which explains the term of a "rule-based parts list" (cf. [Brändl95]).

Other terms used are complex, maximum or selection parts list (cf. [Grupp89]).

The rule-based parts list as quasi standard in automotive industry (cf. [Eisenh02]) serves as implementation basis of modern ERP and PDM systems.

The data model of the rule-based complex parts list is modeled in the application protocol AP214 of ISO 10303 (STEP). It represents the basis for the realization of parts list concepts pursued by various automotive manufacturers.

According to [Eisenh02], Volkswagen AG, for example, documents the vehicle variants based on the rule-based complex parts list in the ES system (*E*ntwicklungs*S*tückliste - engineering parts list).

Final note

Rule-based complex parts lists miss out on a substantial piece of information. They do not directly answer the question of how the individual parts affect each other both in terms of geometry and function and how they are associative to each other[1] (cf. Fig. 2.11).

Position	Level	Part number	Long code rule
POS 100	0	MC101/7/8/9	C1+C3+C5;
POS 101		MC101/03/05	C1+C3+C4;
POS 102		…	…
POS 103	1	MC107/8	C1+C3+C5;
POS 104		MC105/8	C1+C3+C4;
POS 105		…	…
POS 106	2	T107	C1+C3+C5;
POS 107		T105	C1+C3+C4;
POS 108		T106	C1+C2+C5;
POS 109		T104	C1+C2+C4;
POS 110	2	T108	C1+C3;
POS 111		T102	C1+C2;
POS 112	1	MC101/9	C1+C5;
POS 113		MC101/3	C1+C4;
POS 114	2	T101	C1;
POS 115	2	T109	C1+C5;
POS 116		T103	C1+C4;

Fig. 2.11: Example of a rule-based complex parts list
(based on [Herlyn90] and [Ohl00])

Having a closer look at Fig. 2.11, you will notice that part T108, for example, will be used precisely when code "C3" occurs in the vehicle order apart from the (basic) option "C1". Part T102, in contrast, will be used in the vehicle when option "C3" is not contained in the corresponding order. This leads to the conclusion that parts T102 and T108 are alternative parts which exclude each other. The same situation is true for parts T103 and T109. While part T109 is required precisely when the vehicle order contains option "C5" in addition to the basic option "C1", part T103 is used when option "C5" is not wanted. In the same way, parts T104, T105, T106, and T107 mutually exclude each other. Exactly one part of the four alternative parts will be used in any vehicle variant at the same time (cf. [Ohl00]).

Rule-based complex parts lists summarize all parts belonging to a product group in a single parts list. This type of parts list, however, cannot express the reciprocal influence of these parts on each other or their associativity to each other (cf. [Ohl00]).

1 See requirement 3.1.1; 3.1.2; 3.1.3; 3.1.4; 3.1.5; 3.2.1; 3.2.2; 3.2.3; 3.2.4; 3.2.5; 3.2.6; 3.2.7.

2.4.7 Method of complex parts list with ruled positions

For [Ohl00], the decisive breakthrough in mapping products with a large variance has been realized with the **complex parts list with ruled positions**.

This documentation method by [Mann93], also referred to as BCT[1] method divides the complete vehicle in "positions" free of clashes or collisions. The positions document all parts which can be used as alternatives at a specific location and in the same function in the vehicle.

The mutually excluding alternative parts in the same position are called "position variants".

In each position, one position variant can be addressed as maximum for each vehicle order which means that one part or assembly is used as maximum.

The example of a rule-based complex parts list shown in Fig. 2.11 is represented as complex parts list with ruled positions in Fig. 2.12.

Position	Level	Variant	Part number	Long code rule	Short code rule
POS 100	**0**	001	MC101/7/8/9	C1+C3+C5;	C1+C3+C5;
		002	MC101/03/05/8	C1+C3+C4;	C1+C3;
		…	…	…	…
POS 200	**1**	001	MC107/8	C1+C3+C5;	C1+C3+C5;
		002	MC105/8	C1+C3+C4;	C1+C3;
		…	…	…	…
POS 300	**2**	001	T107	C1+C3+C5;	C1+C3+C5;
		002	T105	C1+C3+C4;	C1+C3;
		003	T106	C1+C2+C5;	C1+C5;
		004	T104	C1+C2+C4;	C1;
POS 400	**2**	001	T108	C1+C3;	C1+C3;
		002	T102	C1+C2;	C1;
POS 500	**1**	001	MC101/9	C1+C5;	C1+C5;
		002	MC101/3	C1+C4;	C1;
POS 600	**2**	001	T101	C1;	C1;
POS 700	**2**	001	T109	C1+C5;	C1+C5;
		002	T103	C1+C4;	C1;

Fig. 2.12: Example of a representation of a complex parts list with ruled positions (based on [Herlyn90] and [Ohl00])

As mentioned in the previous section 2.4.6, the parts T102 and T108 or T103 and T109 respectively represent alternative parts which mutually exclude each other. Precisely one part only can be used in a vehicle. This additional information of mutually excluding alternative parts is allowed for in the representation chosen for Fig. 2.12 in as far as the mutually excluding parts are represented as options to be selected alternatively in the respective positions.

1 BCT = Baureihe Code Teil / model range - code - part

Although the information about the mutually excluding alternative parts is also contained implicitly in the representation form of a rule-based complex parts list, the knowledge is used explicitly only when the vehicle is divided completely and collision-free into positions.

The parts assigned in the different positions cannot be combined with each other arbitrarily.

Example: If a customer order contains code "C3", part T108 must be used. This excludes the use of part T102, since parts T102 and T108 are mutually excluding alternative parts. The use of part T108, in turn, excludes the use of part T104 or T106 in position POS 200 at the same time, since these parts are used only when an order does not contain code "C3".

The weakness of the conventional complex parts list in identifying mutually excluding alternative parts just mentioned can be removed with the help of a completely collision-free division of the products into positions.

The consistent application of the BCT method results in a significant advantage for the parts affecting code rules (cf. [Ohl00]). Fig. 2.12 makes clear that the code rules (in the column "Long code rule") contain redundant information with this type of representation. The reason is the bundling of mutually excluding alternative parts.

The structured bundling of alternative parts according to the BCT method allows to express the same information in a shorter way using the code rules "C1" and "C1+C3" (cf. Fig. 2.12, column "Short code rule"). It becomes even more distinct when looking at parts T104, T105, T106, and T107. Here, the long code rules "C1+C2+C4", "C1+C3+C4", "C1+C2+C5", and "C1+C3+C5" are required for an unambiguous control and assignment of the parts. By bundling mutually excluding alternative parts, however, they can be shortened to the (short) code rules "C1", "C1+C3", "C1+C5", and "C1+C3+C5".

When using the short code rules, two requirements have to be observed:

- Short code rules must not be considered isolated, but always in connection with the short code rules of the respective alternative parts. This is due to the fact that the reduction of the code rules is based on a joint consideration of the bundled alternative parts.
- Each position must provide for the position variants for all combinations of codes that affect the respective position. This is absolutely mandatory, since this is the only way to ensure that exactly one position variant will be selected for each position of a customized vehicle to be produced, i.e. one required part or assembly will be selected and used.

To form a long code rule, the short code rule can always be supplemented by the remaining codes of the alternative position variants.

Other advantages offered by the use of the BCT method lie in the context of feasibility checks on the product side. According to [Monke95], a vehicle is feasible precisely when each position has exactly one hit. Hence, an additional feasibility check of the manufacturing orders on the product side becomes superfluous when the BCT approach is applied consistently.

Apart from the shown advantages, the BCT method still holds a decisive disadvantage regarding the complete second level documentation (cf. section 1.1.1). It is not possible to map a possible technical associativity (across positions) or relationship between the parts T102 and T103 for example (cf. Fig. 2.13) with the BCT method. This situation will be detailed in section 2.4.9 (cf. Fig. 2.14 in particular).

Position	Level	Variant	Part number	Long code rule	Short code rule
POS 100	0	001	MC101/7/8/9	C1+C3+C5;	C1+C3+C5;
POS 400	2	001	T108	C1+C3;	C1+C3;
		002	T102	C1+C2;	C1;
POS 700	2	001	T109	C1+C5;	C1+C5;
		002	T103	C1+C4;	C1;

Technical associativity

Fig. 2.13: Problem of missing relation (across positions) of a complex parts list with ruled positions

Final note

Due to the mentioned advantages, the complex parts list with ruled positions is considered to be the most powerful tool for mapping products with great variance currently available. All parts lists mentioned in this study so far can be derived from this documentation type. As quasi standard in automotive industry, this documentation method is widely used in practice in industry. It serves as basis for various IT implementations.

Apart from the advantages, however, the BCT method still holds a few weaker points. The BCT method is still limited to describing the respective products as sum of the parts or assemblies. It is based on the conventional structural relations of the type "belongs to" (component use) or "consists of" (component explosion) which are at the bottom of documenting.

The relations are restricted to mapping hierarchical structures only. For top-down, the components of an assembly can be determined on any level by exploding the components. For bottom-up, the use information of parts or assemblies in superior components can be determined.

The following section will address the weaknesses of this part or assembly oriented structuring method (see Fig. 2.13) and its effects on the various process functions.

2.4.8 Selecting the structuring level

Although the method of open variant parts lists provided for the possibility to document all parts used in a company for any possible product variants in a single parts list, [Ohl00] does not recommend this method. The fact that there are significant differences between the various model ranges, amongst others in the vehicle concepts, makes a bundling across model ranges extremely difficult. However, very many commonalities can be detected within a vehicle model range, even with differing bodywork, which is the reason for [Mann93] to suggest the model range as appropriate structuring level for the automotive industry. The result is a single product structure for each model range which, however, causes the disadvantage that parts used in several vehicle model ranges have to be documented several times leading to redundant data. Besides, it is very difficult to provide experience and knowledge obtained from major components or complex assemblies already in operation for different model ranges[1].

2.4.9 Critical review of current methods

As explained before, the various methods of parts list documentation can exclusively fall back on the conventional structural relations of the type "belongs to" and "consists of".

Although the BCT method uses the feasibility information contained in the code rules regarding the mutual exclusion of parts to be used as alternatives, it does not provide an independent object which allows to manage information between two components on the same structural level[2].

Due to the absence of such an information object, an unambiguous documentation without redundancy for products with great variance is possible to a limited extent only[3].

Information which describes the relation between, e.g., two parts or major components, has to be documented either with redundancy on both "parts" or via an "assembly".

Fig. 2.14 explains this problem using an example. Between the parts T2 and T3 which shall be used together (e.g. for code rule C1), a technical associativity which has to be considered shall exist. The associativity of the two parts cannot be mapped free of redundancy or with additional effort only.

Based on the methods of variant parts lists, this information can only be mapped by

- documenting the two components redundantly (cf. Fig. 2.14 a)
- saving the information in a virtual assembly which has to be documented in addition (cf. Fig. 2.14 b)
- documenting the information in an already existing assembly (on a structural level one level higher as minimum) (cf. Fig. 2.14 c)

---

1 See requirement3.1.8.

2 See requirement 3.1.1; 3.1.2; 3.1.3; 3.2.1; 3.2.2; 3.2.3; 3.2.5; 3.2.6; 3.2.7.

3 See requirement 3.1.5.

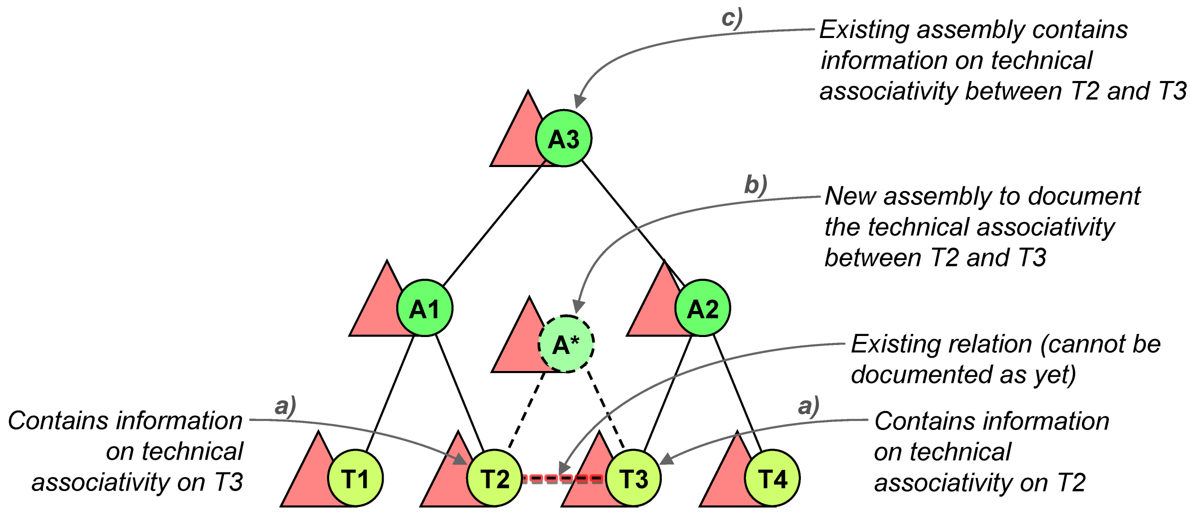


Fig. 2.14: Problem of missing relation between two components on the same structural level

This information to describe the relations can be manifold.

The documenting of explicitly existing **knowledge** based on the hierarchical structural relations in the product structure is feasible to a limited extent only[1].

These structural relations allow for a targeted **variant or complexity management** only to a limited extent. Effects on processes in particular caused by additional part variants can be mapped only very restrictedly. This also means that a reliable management instrument to support decision making is not at hand[2].

The possibility to document **manufacturing relevant feasibility restrictions** unambiguously and without redundancy is missing[3]. The feasibility constraints on the product side, however, can be mapped in the open variant parts lists via the code rules.

In the **parts demand calculation**, the number of parts or assemblies (secondary demand) needed to produce a specific number of vehicles (primary demand) is calculated. Within quantity planning, parts demand calculation is a key factor. Hence, the documentation method of products in parts lists frequently is evaluated and represented by its usability with regards to this parts demand calculation. For a long time, the parts demand calculation for products with a large variance was considered to be especially difficult (cf. [Herlyn90]). The product description based on options solves the general problems. The demand can be calculated for all parts, even in the case of very many product variants. With concrete orders for products, the primary demand is specified in the form of product definitions. The part demand for each individual vehicle can be determined via a direct link of the product to the parts list positions (via the code rules) (e.g. BCT method). Although the introduction of the complex parts list opened up new possibilities and procedures for parts demand calculation, these new

1 See requirement 3.1.5.

2 See requirement 3.2.1.

3 See requirement 3.1.1; 3.1.2; 3.2.3.

procedures are restricted to determine the part or assembly demand. An immediate related demand calculation of required tooling, human resources, and manufacturing times is not possible[1].

A technically complicated and complex product is subject to permanent **changes** during its lifecycle. Due to a sharp competitive situation, legal restrictions and technological progress, the number of changes is especially high in the automotive sector (cf. [Schleg78]). Using the existing hierarchical structural information, an automated determination of all parts or assemblies affected by a change is possible to a restricted extent only. Likewise, the evaluation of the effects and consequences for the manufacturing process using the current data models and methods is also very limited[2].

Along the product engineering process, different **parts list views** are used. To start with, the draft parts list describes the design target (= As Designed) as seen from engineering. In a next step, the design parts list describes the target (= Should Build) as seen from manufacturing. The design parts list captures all components of the product with their technical data from the design point of view. However, this parts list representation can often not be used 1:1 by the subsidiary operative applications, like manufacturing or assembling [SapPlm01]. This is the reason why the parts list positions are captured only in the manufacturing or assembling parts list (= As Built) as seen from manufacturing aspects and assembling states. To describe the operating status in the operational phase, the operational parts list (= As Maintained) exists in addition. This parts list represents the current operating status with respect to service and retrofitting in the operational phase[3]. In practice, other parts list views[4] can be found for the various areas of application. Frequently, the data are not up-to-date since they are retrieved from different parts list views resulting in inconsistent or redundant data.

2.5 Basics and methods of process documentation

The tasks and targets of production planning are of special interest for the conception and development of an integrated product and process documentation. At first, section 2.5.1 gives an overview of the different tasks production planning has to fulfill. Section 2.5.2 will deal with the special meaning of workplans. After general explanations on work planning, this section 2.5.3 describes the special situation of assembling planning. This is followed by the different methods of variant workplans presented in section 2.5.4. In section 2.5.5, reference is made to a computer-aided workplan generation. This part of the study will be concluded with a critical review of the current methods of process documentation in section 2.5.6.

---

1 See requirement 3.1.1; 3.1.2.

2 See requirement 3.1.1; 3.1.2.

3 See requirement 3.2.3; 3.2.5.

4 Examples are disposition parts lists, date parts lists, sales parts lists, warehouse parts lists, calculation parts lists and after sales parts lists.

2.5.1 Tasks of production planning

To ensure that a product can be manufactured in optimum conditions as far as technology, time and business economics are concerned, production planning[1] has to carry out all process oriented engineering and planning activities (cf. [SAP01]). Production planning receives the input data in the form of manufacturing orders, drawings and parts lists[2] (cf. [Schött99]). In the classical organization of business workflow, production planning comprises the following tasks amongst others:

- **Method planning:** The task of method planning is to plan new, improved manufacturing, assembling or testing procedures, and organize the workflow of manufacturing, assembling or testing.
- **Material planning:** The target of material planning is to determine the raw material for the parts to be manufactured at optimized costs and plan the stockholding for the material.
- **Workflow planning and scheduling:** The core tasks of workflow planning and scheduling are to compile workplans and determine allowed times. The substantial input is to consult the engineering and design sectors in designing products in line with manufacturing and assembling requirements and determining the product structure. Further tasks include workplace design and work rating.
- **Cost planning:** To determine the costs of manufacturing the products, the required material, tooling, and wage costs are pre- and postcalculated.
- **Investment planning:** Investment planning for production facilities, machines, fixtures and tools due to product modifications and advancements in manufacturing, assembling and testing procedures represents the key task.
- **Tooling planning:** The main tasks include the planning and development of new tooling as well as their maintenance in tooling monitoring.

2.5.2 Significance of a workplan

According to [Grupp95], workplans belong, together with parts lists, to the most important documents of process documentation. According to [Schött99], a workplan describes

- all actions necessary to convert a rough part or semifinished part into a finished part or
- all actions necessary to assemble an assembly[3]

1 In the following, the terms work planning and process engineering will be used as synonyms to production planning.

2 See requirement 3.1.2; 3.2.3.

3 See requirement 3.1.2; 3.2.7.

Apart from information about the processes to be performed[1] and their sequence, a workplan contains information about the workplaces where the individual processes are performed and the required tooling (cf. [SAP01]). Important work papers, like schedule card, wage and material slip, are derived from a workplan for production planning and controlling (PPS)[2].
[Schött99] distinguishes workplans as follows:

- setup plans
- manufacturing plans
- assembling plans
- test plans

In principle, a (manufacturing) workplan consists of a "header" and a "position section". The header lists information about the entire workplan[3]. The position section is composed of a "material position" and the "operation positions". The material row contains information about the initial material[4] thus providing the link to the parts list. The operation positions give information on the flow of manufacturing[5] where individual work steps can be explained by sketches[6].

At first, the workplan is created irrespective of any order (= basic workplan) and assigned to a product.

2.5.3 Assembling planning

After a rather general look at the tasks and functions of work planning and workplans, the special situation of assembling (planning) shall be addressed in detail in the following.

Manufacturing planning concentrates on single parts, while the planning of assembling operations has to take into consideration that in each assembling operation, several parts are assembled with the means of tools[7] (cf. [Grabow99]). Here, the assembling tasks exceed the mere activities of joining by far. It is not only that the parts must be ready to be assembled, they also have to be stored (intermediate), transported, and possibly be prepared prior to assembling (cf. [Grabow99]).

1 Including their allowed values.

2 See requirement 3.1.1; 3.2.5; 3.2.7.

3 Workplan number, effectivity, up-to-dateness, revision level, author/user data amongst others.

4 Item number, nomenclature, quantity, material type, quantity unit, dimensions amongst others.

5 Operation number and description, cost center, machine group, unit time, wage group, setup time amongst others.

6 See requirement 3.1.1; 3.1.2.

7 See requirement 3.1.1.

Another difference is that in manufacturing planning, a workplan is created for each part to be manufactured, while in assembling planning, separate workplans are required for each assembly or sub-assembly. The final plan in such a sequence is the final assembling plan (cf. [Grabow99]).

Compared to the complexity of manufacturing planning, that of assembling planning is tremendous. In this context, the reader shall be reminded explicitly of the fact that the number of parts in a product group is low compared to the number of the possible assemblies resulting from the combination of these parts. This, in turn, leads to the conclusion that the number of manufacturing plans is low compared to that of assembling plans. The imbalance in numbers is even increased when considering assembling relevant alternatives to assemble an assembly. Such an aspect may be necessary when different toolings (with different levels of automation) are used to assemble the same assembly. In this case, the number of assembling workplans of an assembly is multiplied by the factor of the "differences" in the assembling operations[1].

"Product complexity" directly affects "process complexity" (amongst others the number of assembling workplans to be created and managed). Fig. 2.16 illustrates this situation using the steering unit[2] as example.

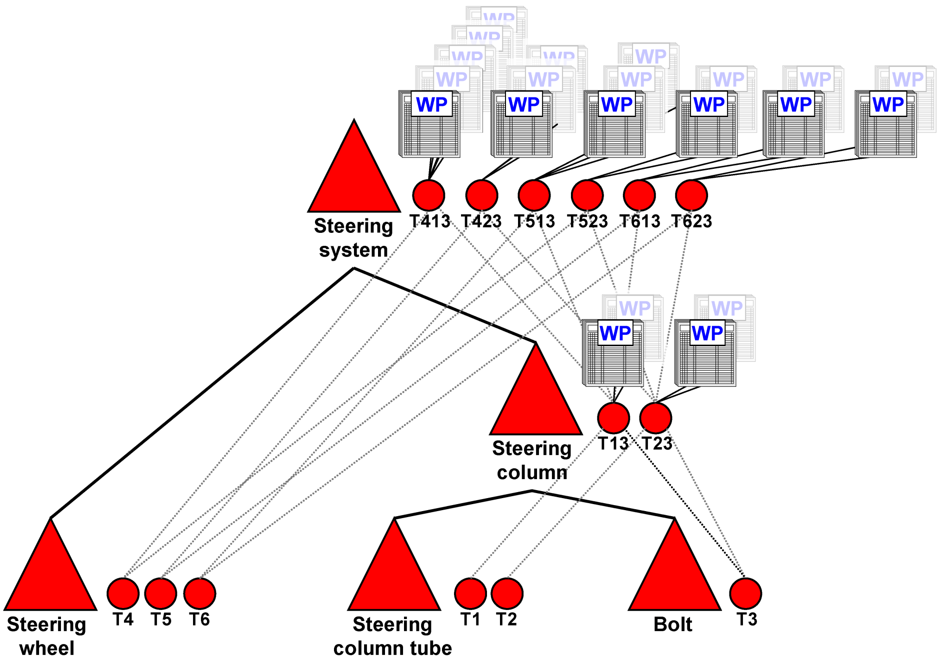


Fig. 2.15: Product and process complexity

In the example, two assembling plans[3] (without allowing for possible assembling alternatives) or four assembling plans (allowing for alternative tooling, e.g.) must be available for the

1 See requirement 3.1.1; 3.1.2; 3.2.7.

2 It is a simplified example without considering feasibility.

3 They result from the combination of the variants for the steering column with the lock. When specifying the number of workplans, no distinction is made on whether they are managed as independent workplans or as basic workplan with reference workplans. Both cases represent assembling workplans. A basic workplan with two reference workplans means two assembling workplans.

"Steering column" assembly. For the next assembling level, six workplans (without allowing for possible assembling alternatives) or twelve workplans (allowing for different assembling times, e.g.) have to be created. Thus, the total effort for planning the assembling of the "Steering unit" includes eight or 16 assembling plans.

For a customer order related manufacturing with numerous variants as it is reality in automotive industry, the generation of assembling plans requires a particularly high effort. And frequently, a reusability of the assembling plans cannot even be granted. Often, assembling work is carried out with the support of manufacturing/assembling parts lists which are supplemented by the relevant assembling notes (cf. [Grupp95]).

The problems of different parts lists and parts list views were already addressed in section 2.4.9 and shall not be advanced here.

2.5.4 Variant workplans

Variant workplans have the same motivation as variant parts lists. If a number of workplans are identical except for a few positions, it is sufficient to document the identical operation positions in a basic workplan and list the deviating operations in referencing plans (= reference workplan).

As for the variant parts lists, there are four general combinations of basic workplan and reference workplans possible for the variant workplans (cf. [Grupp95]):

- Creating a shrinked workplan which contains all operations of a part series that remain the same. Besides, there are part specific variant plans (cf. [SAP01]). This type of variant planning corresponds to the concept of a common parts/variant parts list described in section 2.4.4.
- Creating a complete basic workplan plus corresponding variant workplans. The variant workplans contain the plus-minus work processes deviating from the basic workplan. This type of variant planning corresponds to the concept of the basic version and plus-minus parts list described in section 2.4.4.
- Creating an operation record with several time fields for each part variant to mention an example. Besides, it might be necessary to provide for separate fields for fixtures. The basic workplan comprises all parts of a part family. This type of variant planning corresponds to the concept of a type parts list described in section 2.4.4.
- Creating a joint basic workplan where the differences are specified in tables. In relation to the customer specific product definition, the basic workplan is supplemented by the respective operations from the additional tables to produce the ordered product (= "selection workplan" or "combinable workplan"). This type of variant planning corresponds to the concept of the (open) variant parts lists described in section 2.4.5.

2.5.5 Computer-aided workplan generation

Computer-aided planning (CAP) or computer-aided process planning (CAPP) are frequently used terms in this context (cf. [Schött99]).

Compiling workplans is the most time consuming part of all work planning activities (cf. [Grabow99]). Therefore, the first step target of computer-aided planning is to support the compiling of workplans by using the available data. The following activities are distinguished[1]:

- Filing workplans by an explicit saving of individual workplans including all related data.
- The managing of references where the operation contents are stored centrally once. The link to a concrete workplan is realized via references.
- Filing an algorithm to create workplans. Starting from a part description, a computer can generate a workplan automatically. Subsequently, the workplans generated in this way can be managed in an ERP system, for example.

As with the design types, the generation of workplans[2] is distinguished by the planning types (cf. [Grabow99]):

- Adjusted planning: This type of planning is based on the modification of saved workplans. Workplans are generated with the following steps:
  - Search a workplan which is similar to the current planning operation
  - Copy the existing and usable workplan
  - Modify the copied workplan
  - File the new workplan
- Similar planning: This planning type, referred to as variant planning (principle of variants) is based on a class of parts which are similar in terms of geometry[3] and manufacturing technique[4]. Planning a variant starts from a "standard workplan" as planning foundation.
- New planning: With this type of planning, which is also called workplan generation, similar, already filed workplans cannot be used. This type centers on algorithms which generate all workplan data based on the input data. This type of workplan generation is mainly used to create manufacturing workplans[5].

---

1 See requirement 3.1.6; 3.2.7.

2 See requirement 3.1.5; 3.1.6.

3 Similar geometry means that the shape and dimensions of the parts can vary within defined limits.

4 Similar manufacturing technique means that the operations are identical or show minor deviations only.

5 See requirement 3.1.6.

Manifold methods exist to support the computer-aided generation of workplans. At this point, group technology (cf. [Grabow99]), feature technology (cf. [Haasis95]) and the methods of artificial intelligence (et al. expert systems, fuzzy logic and case-based reasoning) (cf. [Heinri96]) shall be mentioned as examples. In the context of work planning, these technologies are mainly used in manufacturing planning today. Regarding assembling planning, these approaches are less important. In the work planning field, the primary aim of these technologies is not to integrate the input data as is the target of the interconnections documentation.

2.5.6 Critical review of current methods

In corporate workflow organization, the tasks of design regarding product engineering and those of work planning for process engineering are still being realized in a sequential, work dividing way in most cases (cf. [Bochtl96]). This sequential procedure calls for revisions in case the product design does not live up to manufacturing demands. The resulting iteration loops mean a loss in time and increase in development costs. Frequently, the interface between engineering/design and work planning is restricted to the exchange of CAD models, drawings, and parts lists (cf. [Schött99]). Typically, these descriptions lack the logical correlation which would allow to recognize manufacturing elements as well as the sequencing of assembling steps (cf. [Bruhn01]). Hence, the generation of the workplan documents requires much effort and time (cf. [Grabow99]). The workplan represents the database for all functions which are related to the sequential flow of a manufacturing order. As interface between work planning and production planning and controlling (PPS), the workplan is of very special significance. Although this fact places work planning in a central position between engineering/design and manufacturing (cf. [Bruhn01]), today's methods and IT implementations do not adequately support this position. In the start-up phase especially, this has particularly striking effects which become manifest in an increased number of modifications[1].

Errors which are not recognized during product engineering, work planning or parts manufacturing cause significant efforts in assembling. To avoid such errors or to address these problems early, numerous methods for an assembling adequate product design (design for assembly, DFA) are used in practice (cf. [Grabow99]). Besides, auxiliary means, like the preference graph showing the operation sequence as single operations in a graphical form, are used (cf. [Grabow99]). IT solutions which integrate product and process information to the maximum and support an automated workplan generation for assembling are not available, which is all the more true for manufacturers of series with large variance[2].

1 See requirement 3.1.1; 3.1.2; 3.2.3; 3.2.4; 3.2.5; 3.2.6; 3.2.7.

2 See requirement 3.1.5; 3.1.6; 3.2.5; 3.2.7.

2.6 Implementations of product and process documentation

This section presents various solutions of an integrated product and process documentation as examples. The solutions presented in sections 2.6.1 to 2.6.3 are implementations which are proven or currently used in practice. These implementations are based on the methods of product and process documentation presented in sections 2.4 and 2.5. To supplement this, section 2.6.4 presents different scientific concepts which are implemented as prototypes.

2.6.1 Documentation at Mercedes-Benz Cars as example

In the field of *Technical Documentation*, different methods and IT solutions exist for the different business units of Daimler AG. Starting from the core processes at Mercedes-Benz Cars (MBC), this section will introduce the methodological concepts and essential components of product documentation at MBC.

The **product generation process** and the **customer order process** form the core processes (cf. Fig. 1.3) at MBC.

The product generation process (= PGP) forms the authoritative basis for working in the engineering process across business units at MBC. Control points, the "quality gates", lead to a higher product and process maturity level due to binding customer-supplier agreements, benchmarks and well directed measures. The quality gates are used to make authoritative and consistent decisions on the further procedure in the process based on the demands to be met. This creates a reliable basis for the future work which is relevant for all parties involved. In this context, product documentation plays a central role. The parts list created in product documentation ensures a transparent engineering status in all fields involved by providing data at an early stage. It thus represents a generally valid and uniform database for

- the planning fields of engineering
- logistics
- production
- material purchasing
- cost calculation

The first parts list draft is created in coordination between engineering and product documentation using information from the performance and requirement specifications as well as additional information from the design units. In the course of the engineering process, the parts list is completed gradually until it reaches a series close engineering level. At this point in time, drawings or geometry data exist and are documented in the parts list. In coordination with the design units, activities for an authoritative production means procurement can be initiated.

When a part is ready to enter series and released by the design unit without limitations, the production sectors can trigger all measures required for material and production tooling procurement as well as the sampling process for the series-startup (cf. [Alam02]).

Next to the product generation process, the customer order process (COP) is the second core process at MBC. The complete fulfillment of the quality gates in the product generation process forms the prerequisite for the realization of the customer order process.

When the product configuration has been completed by defining the customized vehicle properties, the salesperson enters the customer order into the system[1]. An order related feasibility check is performed for each customer order which consists of several steps according to the product complexity. If there are no conflicts, the dispatching of the order into the order backlog of the envisaged manufacturing plant is triggered. The order dispatching integrates the customer orders received during a day into the "string of pearls"[2]. Production order controlling converts the vehicle orders scheduled when creating the string of pearls into production orders which are distributed to the shifts of a day. Demand calculation uses the product documentation and the active production orders to determine the corresponding part demand. The demands are converted to delivery schedules which are provided to the suppliers. Delivery is realized via different procurement chains[3]. In the next step, the vehicle is actually built. Picking, inspection and assembling of the parts is realized in the assembling locations for the body-in-white or the assembling stations. After the final approval, the vehicle with the final approval report and vehicle documentation is transferred to shipping (cf. [Christ97]).

Product documentation as the element to link the two core processes forms the basis for all corporate processes at MBC. As a component of the ERP system, is consists of the product overview and the parts list.

The **product overview**[4] documents all technically feasible and sales released vehicle and equipment variants and available optional equipment (cf. [Matt04]). Fig. 2.16 shows the logical data model of the product overview.

1 Depending on the customer order, the basic vehicle data (et al. model range, variant, type, model), information on optional equipment and the desired delivery data are included.

2 Different criteria, like e.g. production capacity, since additional capacity limits have to be observed also in combination for some optional equipment (e.g. sunroof or air conditioning), or color blocks for painting, have to be observed.

3 Sequence related procurement chain, order related procurement chain, program related procurement chain (order/consumption controlled) or consumption related procurement chain.

4 The product overview corresponds to the product level described in section 2.5.

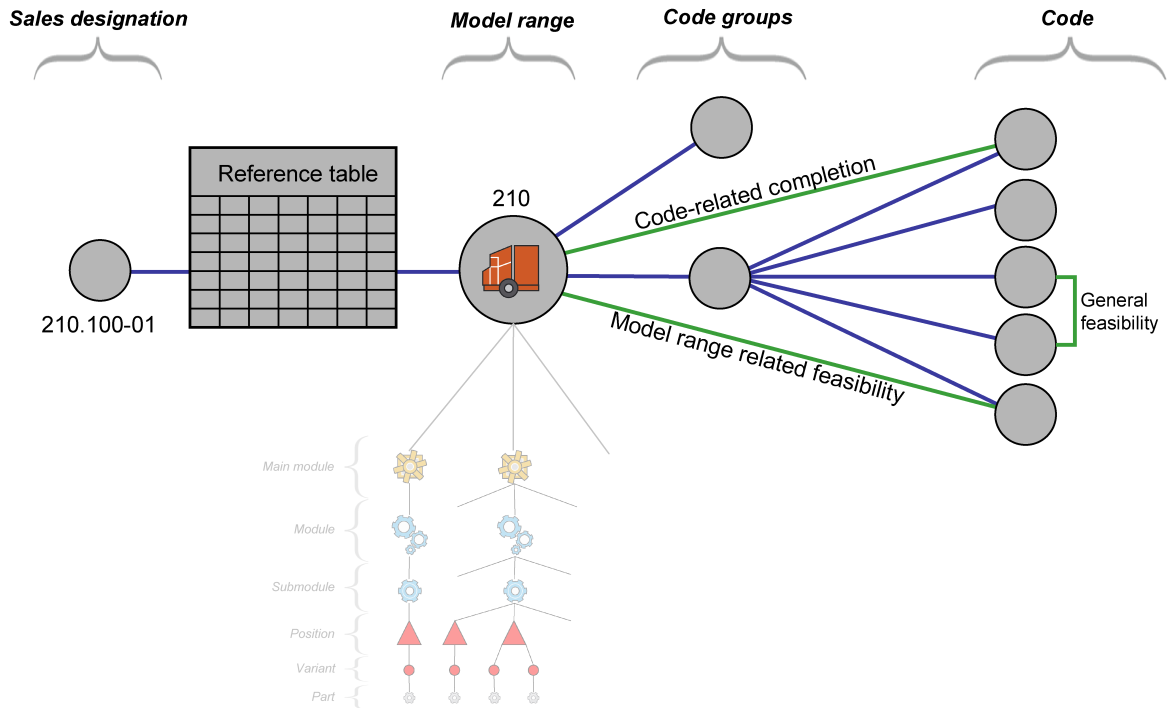


Fig. 2.16: Logical data model of product overview

While models represent the sales designations used by the sales department for customer communication, both product overview and parts list are documented in relation to model ranges. Here, the model reference in the form of a reference table (cf. Fig. 2.16) serves as internal implementation of the sales language into the documentation language.

The link to the parts list is ensured via codes. Codes mean the coding of sales terms (optional equipment, options) created under the aspects of sales, functions and controlling. Codes can be ordered by the customer or completed inevitably[1]. They are required for an unambiguous determination of the vehicle variant. In the parts list, they are used to determine the parts. Consequently, order and parts specifications are linked together closely.

Due to reasons of clarity, codes are not assigned directly to the model ranges, but to code groups. Code groups are used to bundle identical or similar codes of a model range (cf. [Alam02]).

The feasibility check for a customer order is performed based on the feasibility rules stored in the product overview. It is checked whether the codes contained in the order are technically available and allowed for the order. It is distinguished between the

- general feasibility and the
- model range related feasibility

1 A general distinction is made between the code related code completion (if a code requires others codes to include all parts in the parts list required for the assembling) and the model range related code completion (if a parts list comprises codes which are documented in the parts list, but not transmitted explicitly by the salesperson) (cf. Fig. 2.16).

The general feasibility checks the exclusion of one code to another code. It applies across model ranges for all model ranges where these codes are valid. The model range related feasibility determines the availability or exclusion of a code in relation to another code (cf. Fig. 2.16). This applies to the model range this code is valid for.

The **product structure/parts list** at MBC is based on the BCT method described in section 2.4.7.

The top entry term for a parts list is the model range. It is an umbrella term for vehicles or major components of the same type. In model ranges, all feasible vehicles (of a vehicle model range) or major component variants (of a major component model range) are documented in the parts list (cf. [Alam02]).

Fig. 2.17 shows the logical data model of the product structure.

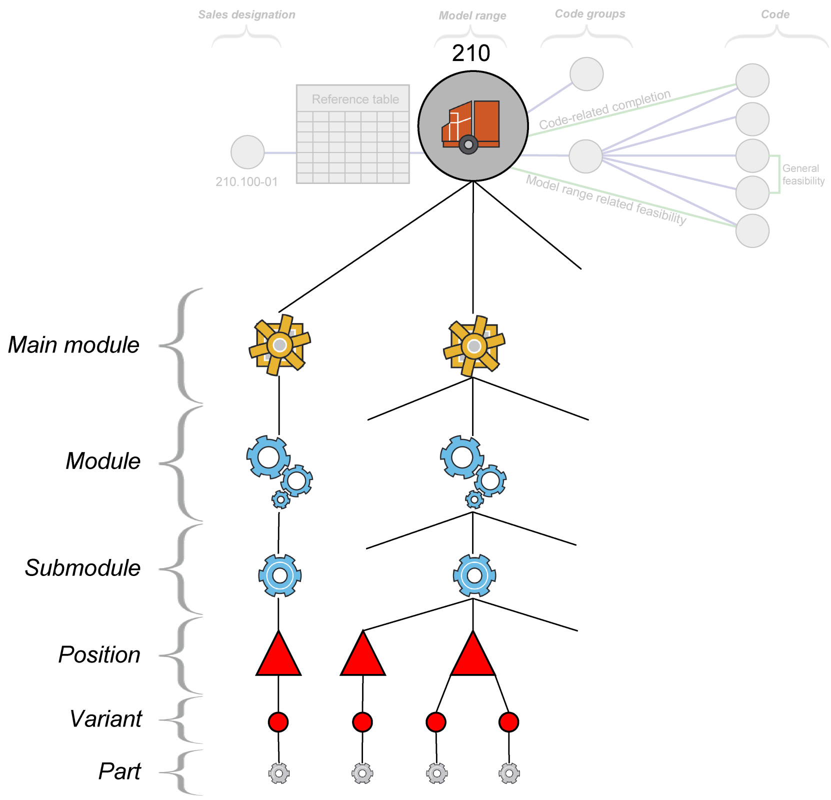


Fig. 2.17: Logical data model of product structure based on [Matt04]

Due to reasons of clarity and to allow for an easier retrieval of information, a generally valid and standardized parts list structure was defined for all model ranges. This structure consists of a three-level module grid which is built according to functional and geometric installation constraints (cf. [Merced95b]). The module grid is divided into main module, module and submodule (cf. Fig. 2.18). Below the submodule level, the parts list structures are documented via parts list positions which, in turn, are structured hierarchically.

Parts list positions document all parts which can be used as alternatives at a specific location and in the same function in the vehicle. In general, there is a single level relation from the model range to the part position. The alternative parts in the same position are called "position variants". Using code rules (and effectivity dates), the conditions when the parts shall be used are documented. Starting from the total parts list, the parts lists for each individual order can be determined precisely.

The relation between geometry and parts list is of special significance. Since geometry and parts list do not emerge independently of each other, a close link between the geometry managing EDM system and the parts list system is required. In principle, two different methods to link geometry and parts list are possible:

- The ERP/EDM method comprises the generation of parts list structures in the parts list system with a subsequent supply to the EDM system. In the EDM system, the data are supplemented by geometry oriented structures and returned to the parts list system (cf. [Buchte02]).
- The EDM/ERP method is an advancement of the ERP/EDM method. The engineering process starts by building the complete product structure in the EDM system. Afterwards, the product structure is transferred to the parts list system where production specific information is added. This facilitates the data stream between EDM and parts list system.

To realize **process documentation** in the product generation process, an integrated planning system (iPS) is used at MBC. The conceptual objective of this planning system is the creation of an integral, up-to-date database. The core of this database is the parts list to which all plans[1] are linked.

In the iPS workplan, the workflow is planned, and manufacturing workflows, times and means are documented. For the producing sections of press shop, body-in-white, surface coating and assembling, the workplans are carried out in an appropriately modified form. There are two types of workplans:

- Design workplan: This type of plan is used for the parts manufacturing, and describes how the parts are shaped from raw materials.
- Assembling workplan: This type of plan is defined for the assembling process. In an assembling workplan, all activities required for the assembly line process are planned and documented in principle. Assembling plans are divided into basic and additional plans (cf. section 2.5.4). A basic plan contains all operations which require the same time for the respective work steps for each vehicle reference type. Basic plans are always valid for the entire model range. Additional plans are created for each specific vehicle reference type and contain the extra time needed for the respective reference type. The additional plans contain the information for optional equipment and parts with code rules.

1 Apart from the workplan, this includes logistics plan, resource demand plan, material purchasing plan and precalculation.

Workplans in iPS contain several operations. At first, the operations are documented irrespective of their manufacturing sequence and assigned to a work area. Subsequently, the manufacturing sequence is determined in two steps. In the first step, the sequence of the work areas is determined. In the second, the sequence of the operations within the individual work areas is defined.

The parts list is integrated into the work planning by linking the planning data to the parts list on the level of the position variants (cf. also Fig. 2.15). There are two types of linking:

- A workplan is linked directly to a position variant. In this case, the workplan is selected via the code rule of the position variant.
- An operation is linked to a position variant and thus to a concrete part.

Plans which cannot be assigned to a position variant of design (wash vehicle, for example) are linked via a submodule especially reserved for production. It is in this submodule exclusively that planners can define own positions or position variants (cf. [Sheikh03]).

2.6.2 Standard for the exchange of product model data (STEP)

The key basis for using integrated product models in industry is the "Standard for the Exchange of Product Model Data" (STEP, ISO 10303) (cf. [Anderl94], [Grabow94a], [Grabow94b]). In the STEP data model, the different methods of product and process documentation described in the sections 2.4 and 2.5 can be mapped "free of losses". Hence, the STEP data model maps both the advantages of the different documentation forms and their restrictions.

STEP in general summarizes the product data models of the standard series ISO 10303. For the description of product reference models in STEP, the data modeling language EXPRESS and its graphical subset EXPRESS-G are used. EXPRESS is object oriented, but puts emphasis on the definition of data structures. Although there are elements to define rules of the data objects among each other, they are not suited for the description of complex processes. Methods are not described at all. Therefore, the particularly weak points of EXPRESS, and hence STEP, lie in the mapping of process flows. Process states, however, can be mapped and exchanged without problems.

The scope of ISO 10303-214[1] comprises in general the engineering process for mechanical components and the necessary tools in automotive industry and suppliers[2]. As is typical for PDM and CAD systems, the application protocol AP214 starts from a hierarchical product structuring with a splitting into a product, its assemblies and single parts which can be

1 ISO 10303-214: Industrial automation systems and integration - Product data representation and exchange - Part 214: Application protocol: Core data for automotive mechanical design processes.

2 Electrics, electronics and hydraulic and pneumatic components are ignored explicitly here (refer to ISO 10303-212).

mapped with relatively simple constructs. It has to be distinguished whether configuration management is applied or not.

Without configuration management, the products and assemblies are mapped to the objects "item" and "item version" in AP214. On this level, however, code rules are not applied. Here, data exchange always means the exchange of the parts list of a specific instance.

When a configuration management is applied, data objects must be used which are described in AP214, Units of Functionality S7. A product structure which allows for the different configurations is built by defining rules for the individual positions in relation to the product class which allow for the selection of the correct part (see Fig. 2.18). Basically, AP214 thus describes the procedure related to the BCT method.

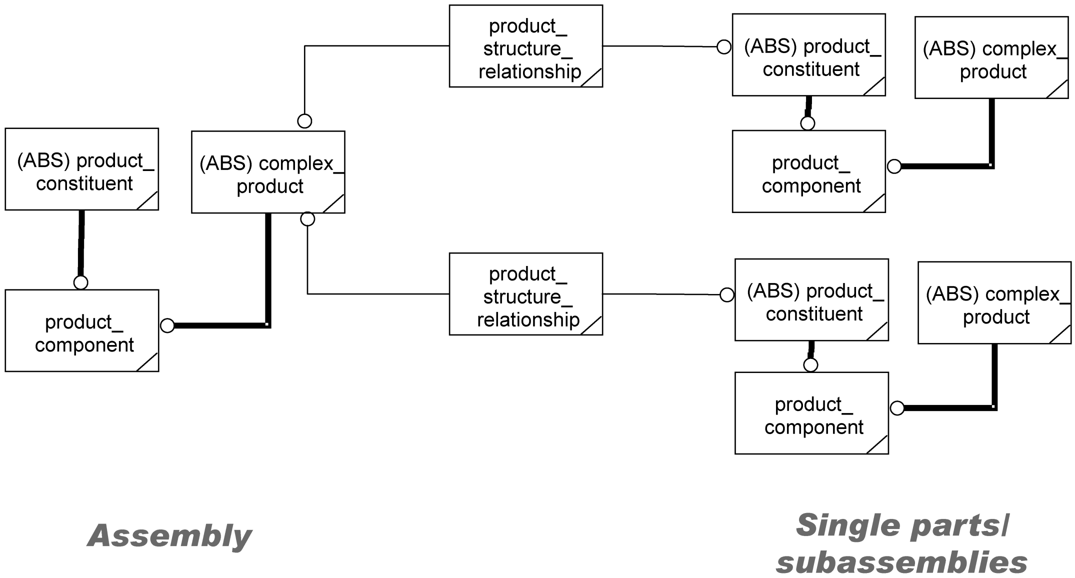


Fig. 2.18: Product structure in AP214
(simplified representation)

2.6.3 Commercial application systems using SAP as example

To support the integrated product generation process for series products with large variance which are customer configured and manufactured on cycled assembly lines, the *integrated product and process engineering* (iPPE) was developed within the mySAP.com Automotive initiative.

The iPPE consists of the following application object types:

- The application object type CMP represents the product structure. The product structure describes the structure of a product.
- The application object type ACT describes the process structure and the workplan in iPPE. The process structure is used to describe and structure the activities required to manufacture the product.

- The application object type FLO represents fabric layout and line design. The line design is used to represent the layout of production facilities.

The application object types can have assigned various elements (nodes, variants, alternatives and relations). The assignment of the elements to a specific application object type defines the task an element assumes in iPPE.

Relations are used to link the elements of iPPE. Each relation links a pair of elements. There are different relation types:

- A relation runs from the superior node to the subordinate node representing an "is part of" relation
- A relation runs between two successive nodes representing a sequence or "sequence relation"
- A relation runs between two nodes of different applications (product structure, process structure and line design) defining an assignment

Fig. 2.19 shows the application object types and their relations:

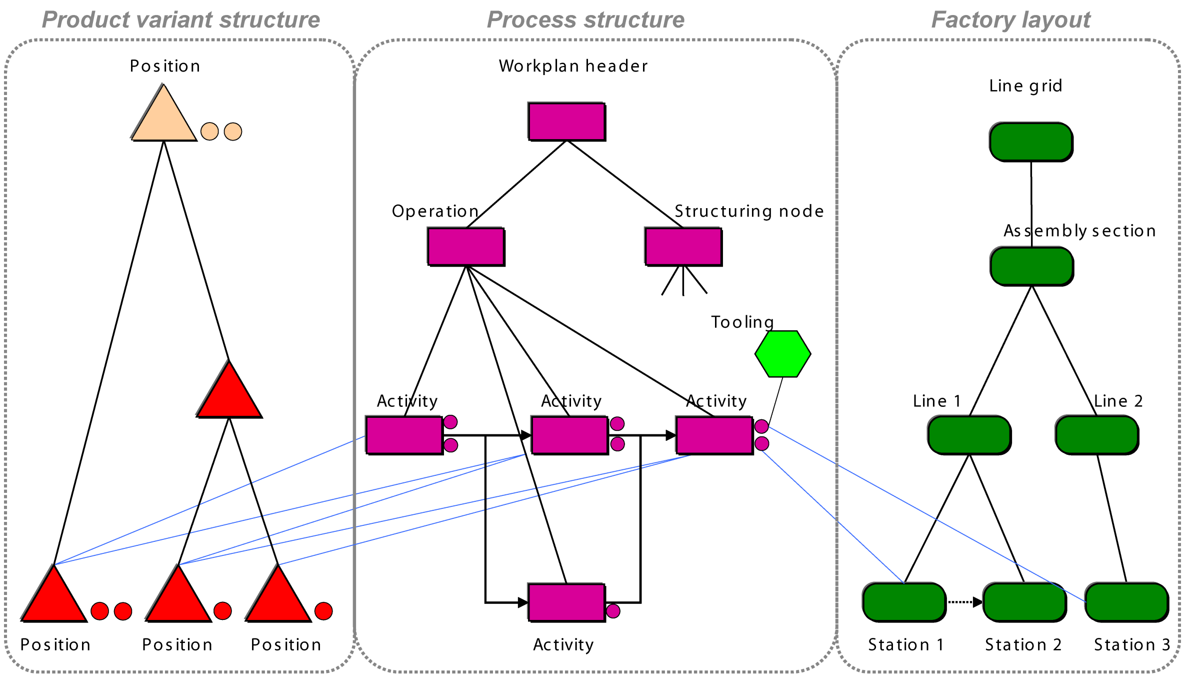


Fig. 2.19: Schematic representation of application objects within iPPE

The **product structure** of a complex product or product family is mapped via a product variant structure (PVS) (cf. Fig. 2.19) which was developed based on the STEP AP214 standard (cf. section 2.6.2). In the PVS, the product structure can be built considering functional aspects to be expanded step by step according to the progress in engineering. To build the PVS, nodes of the type "structuring node" are available. Structuring nodes represent functions of a product whose concrete values are the variants. Variants (position variants) at structuring nodes can have one or several materials assigned. Relation information can be stored at the variants of the product structure. This information is used to control the selection of the variants at the structuring nodes. Thus, PVS corresponds in terms of methodology to the complex parts list with ruled positions described in section 2.4.7.

Like a classical workplan, the **process structure** describes the manufacturing flow to produce own materials/products. It consists of four different node types which are used to map the workplan data (cf. Fig. 2.19):

- The node type "workplan header" forms the top level of the process structure. It defines the sequence of the activities to be performed thus corresponding to the workplan header of the classical workplan.
- Activities represent the basic element of the process structure describing how a work step has to be performed. The mode specifies how and where the activity shall be performed. The components to be processed with the respective work step are assigned to the activity. In addition, the activity specifies the subsequent activity. Via the upward hierarchical relation, the activity is assigned to the superior structuring nodes and the workplan headers. Hence, an activity corresponds to an operation in the classical workplan.
- Structuring nodes can be used to bundle several activities and operations which occur repeatedly in the same sequence for a multiple use. These structuring nodes can then be inserted in different process structures. Apart from the activities, structuring nodes can also be assigned other structuring nodes building a multilevel structure. Structuring nodes correspond to the standard workplan of a classical workplan.
- The node type "operation" is used to bundle activities. However, it is not possible to map a multilevel structure.

The **factory layout** or **line structure** is used to map production resources for cycled line and series manufacturing. A line structure consists of different line elements with specific functions which can be arranged across several levels in a hierarchy. Within the levels, the elements can be arranged in a sequential order to form a grid. The following line elements are available to structure the factory layout (cf. Fig. 2.19):

- A line grid is used to bundle several lines to a unit.
- A line is the element to be planned within the line structure the respective production resources are assigned to.
- Buffers can be inserted between two lines which allow for an intermediate storing of orders to bridge logistic problems between the lines to mention an example.
- In contrast to lines, line sections do not represent independent resources which can have plan data, but serve the more detailed structuring of a line. Line sections can be divided further across any number of hierarchical levels. Within these hierarchical levels, sequential relations are used to build a grid-type structure of the line sections.

- Work areas separate a line section or line. Work areas are used to map different installation locations within a line section.

2.6.4 Scientific approaches and prototype implementations

The previous sections 2.6.1 to 2.6.3 described different implementations of an integrated product and process documentation which are used in practice in industry. The following section will address different representative scientific approaches as examples.

Integrated product and production data model (PPM)

The concept of the PPM by [Grabow95] comprises an object oriented data model (object model) for a computer aided processing and managing of data which are required in engineering and product generation. The key advantage of PPM as integrated model is the fact that it describes a product or production tool in all its lifecycle phases in a consistent way from a central point of view. The aim is to design the PPM as a concept for the data oriented integration of engineering applications which can be used to realize a continuous flow of information across the different applications. To achieve this, a core structure was developed in PPM for a general definition of physical, matter based objects. Fig. 2.20 shows the relevant object types of this core structure.

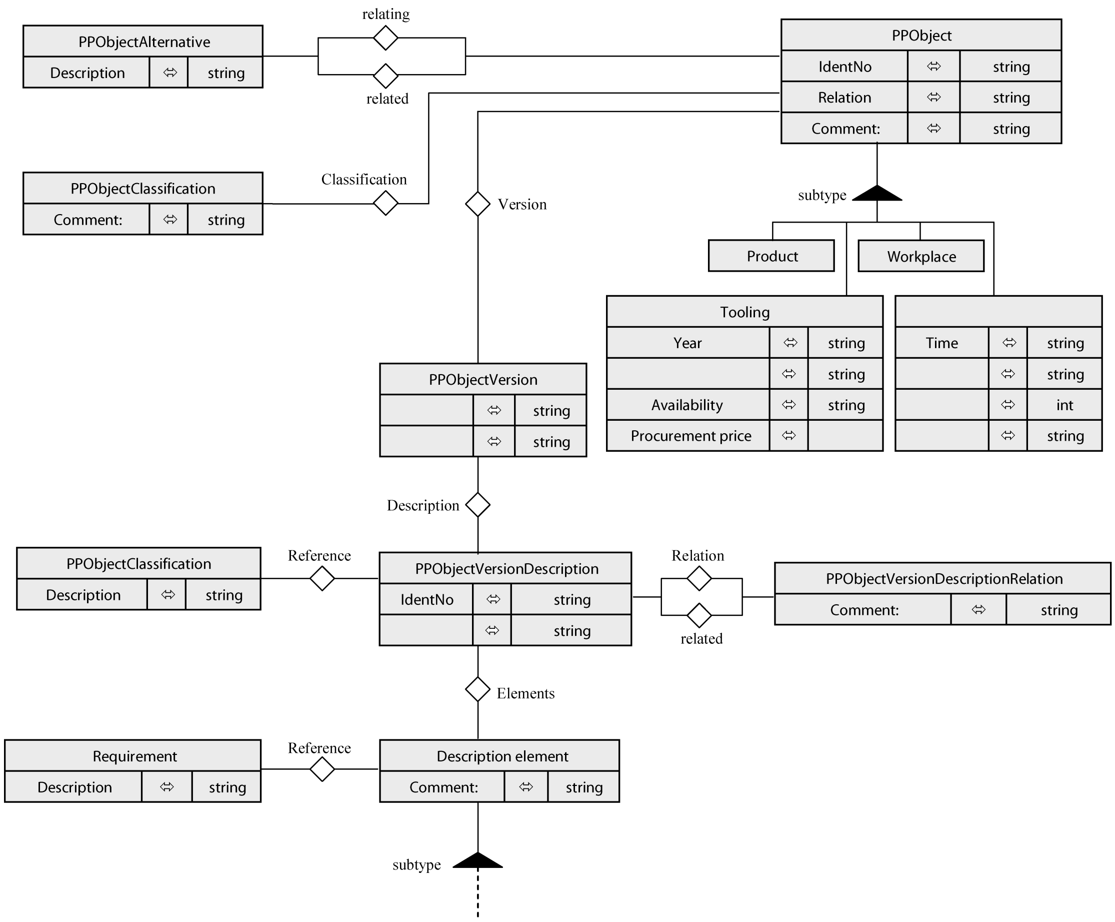


Fig. 2.20: Core structure of PPM
(based on [Grabow95])

Within this core structure, the object type PPObject represents the central object type. Real objects are detailings of this object type. Apart from the selling products and the tooling required for their manufacturing, individuals and workplaces are described. It is envisaged to map products with large variance including their structural layout as described in section 2.4 as well as integrate process information.

Integrated modeling of process and product in engineering

The center of this approach according to [Ott97]) is the IT technical integration of product and process models in product engineering. A two-level model is used to distinguish the information in a generally valid description level (schema level) and business case specific values (instance level). The schema is described in a formal (e.g. EXPRESS) or non-formal language (e.g. English). The instantiated model is the value of the schema for the individual use case.

It is possible to link product and process model on the instance level. With an integration on the schema level, links between product and process model have to be defined already in the schema. The granularity of the links on the schema and instance levels is considered to be an important aspect of integration. In relation to the granularity, sensible and less sensible links between product and process model are distinguished.

A formal description of the processes by an information model is the prerequisite of mapping the instantiated product engineering processes. For the mapping of the instantiated product models, it is recommended to start from independent tools and standards (ISO 10303 AP214, cf. section 2.6.2). As a result, product modeling is subject to the limitations for products with large variance described in section 2.4. Due to a "separate" product and process modeling (cf. section 2.5), a true integration is very difficult to achieve.

Continuous support of B2B value chains

The focus of investigation for a continuous support of value chains according to [Gausem00] is limited to the process chain (technical marketing, project planning, order processing, post sales support) on the supplier side.

The central core of building a process chain supported by IT is an object oriented information model (information modeler). The information modeler assumes the task of storing and managing all information which occurs in the process chain. Apart from product information (technical parameters, CAD models, etc.), this includes information on the structure and search functions. The decisive property of the information modeler is that the different contents and classifications are the constituents of a shared overall information structure. A generic meta model allows for a dynamic expansion. No programing effort is required to store or manage new information structures. A feature to map products which large variance for series manufacturers, as is the case in automotive industry, is not envisaged explicitly. In addition, a methodological procedure of how to provide a corporate management tool based on an integrated product and process information model is not described.

Product engineering which lives up to logistics demands by linking product and production modeling

Targets of linking product and production models according to [Bley97] are:

- planning the production system
- design in line with logistics demands
- cost calculation

The investigation concentrates on the integration of product data from engineering/design into simulation capable material flow models. This offers the engineering sector a tool which allows to obtain information regarding the consequences of a concrete product design and investigate possible design alternatives. For the linking of the data, the use of an EDM system or product databases is recommended. For mapping products with large variance or integrating product and process information, this approach is subject to the restrictions described in section 2.4.9.

Variant management is based on the approach of variance avoidance (cf. section 1.1.1). Considering the *variant mode and effects analysis* (VMEA), the developed approach can be used to perform a cost/benefit analysis to clarify the extent to which a new variant justifies anticipated extra costs.

Integrated product and process modeling

According to [Faux94], the focus of this integration is on the product engineering process. The product model is supplemented by a process model which defines all design and work preparing phases. Based on features, a semantic product modeling is performed. The process modeling component is integrated to a rudimentary extent only.

Part II

REQUIREMENTS

3 Requirements on an integrated product and process documentation

Based on the analysis of current product and process modeling methods performed in chapter 2 this part formulates the requirements placed on an integrated product and process documentation. This study, however, cannot provide detailed requirement specifications in the sense of technical and requirements specifications. A (detailed) specification of the interconnection based product and process documentation would go far beyond the scope of this work. This part shall address the key requirements and criteria both for an integrated product and process documentation and its IT technical realization.

Considering the corporate wide targets regarding cost, time and quality, a division of the requirements into the categories

- variant management (see section 3.1)
- process design/management (see section 3.2)
- IT systems or IT technology (see section 3.3)

appears to be sensible following the situation of manufacturing companies or the automotive industry as described in section 1.1.

Section 3.4 shows a summary of the requirements placed on an integrated product and process documentation. For the respective requirements, reference is made to their conceptional realization in chapter 4 and their prototype implementation in chapter 6.

3.1 Requirements as seen from variant management

This section describes the requirements placed on an integrated product and process documentation as seen from variant management. The method of *Technical Documentation* is seen as an indicator both for the complete mapping of all theoretical product variants (first level completeness, cf. section 1.1.1) and the complete mapping of product, process and resource information of a product (second level completeness, cf. section 1.1.1).

3.1.1 Integration of product, process and resource information

Conventional product structures describe a product as the sum of its parts or assemblies. As structuring means to describe the products, relations of the type "consists of" or "is used by" are available exclusively. The product structuring methods have weaknesses in this context.

The aim is to realize a (vertical) continuity from the entire vehicle (as seen from sales) down to the parts level (as seen from engineering). An immediate related demand calculation of required tooling, human resources, and manufacturing times is not possible. The method does not plan to consider the process information. A (horizontal) integration of process information is only possible in a very rudimentary form. More than ever, product structures must be capable of mapping webs which consist of parts and their interrelations. It is less the number of different parts in a vehicle than their interrelations which is of importance. It must be possible to map the technical associativity of parts and assemblies including their manufacturing and assembling information. The product structure must be capable of living up to the demand of being the linking element between product and process. Data models and methods must be suited to map effects on the processes and consequences for the manufacturing process. It must be possible to document manufacturing relevant feasibility limitations unambiguously and free of redundancies. It must be possible to consider feasibility restrictions on the production side for a communication between manufacturer and customer in the product structure without having to document redundant structures. Future oriented product structures have to allow for a methodological consideration of process and resource information obtained during the product generation process. The recognition of logical correlations of elements regarding manufacturing techniques and the sequence of assembling steps must be supported. Information about the operations to be performed, their sequence and workplaces where the individual processes are performed as well as the required tooling has to be managed. The prerequisite of an integrated product and process documentation is to consider the product, process and resource information in a logical data model (cf. 1.1.1; 1.1.2; 1.1.3; 2.2.4; 2.2.8; 2.4.8; 2.4.9; 2.5.2; 2.5.3; 2.5.6).

3.1.2 Integration of product engineering and production planning

Together with the parts list, the workplan is the most important document of production planning. Nevertheless, the evaluation of the effects and consequences for the manufacturing process based on the current data models and methods is very limited. However, it must be possible to consider assembling relevant alternatives for producing an assembly. Such an aspect may be necessary when different toolings (with different levels of automation) are used to assemble the same assembly. The material row contains information about the initial material thus providing the link to the parts list.

Starting from a part description, it must be possible to generate workplans via computer programs. These workplans must be available for the downstream processes. It must be possible to document the manufacturing progress of the products and their parts and components.

Parts lists and their assemblies have to master the aspects of manufacturing technology and possess a reference to manufacturing and assembling. A horizontal integration of the process information shall be realized. The close link between product and process engineering is absolutely mandatory (cf. 1.1.1; 1.1.3; 2.2.8; 2.4.4; 2.4.8; 2.4.9; 2.5; 2.5.1; 2.5.2; 2.5.3; 2.5.5).

3.1.3 Flexible and easily expandable data model

The data model of an integrated product and process documentation must be flexible and easy to expand. An independent object must be available to manage information between two components of the same structural level. It shall be possible to consider new objects or attributes in the respective updated data model with minimum effort (cf. 2.4.8; 2.4.9).

3.1.4 Unambiguous and redundant free mapping of information

The splitting into the two core processes "customer order processing" and "product generation" and the related separation into commercial and technical information systems results in a heterogeneous system world and a variety of system interfaces. Redundant, inconsistent or outdated data as well as poor data quality are the consequence. Often, they lead to instable processes. The functional separation of the sectors with the trend to their internal suboptimizing increases friction and information losses.

While an unambiguous and redundant free documentation for products with large variance based on the conventional structures is possible to only a limited extent, future product structures must be able to map webs of parts and their associativity to each other. It must be possible to map attributes unambiguously. Redundant data between the individual parts lists must be avoided. The documentation of the product, process and resource data must be unambiguous and free of redundancy (cf. 1.1.1; 1.1.3; 1.1.2; 2.2.8; 2.4.2; 2.4.8; 2.4.9).

3.1.5 Documenting experience and knowledge

The documenting of explicitly existing knowledge based on the hierarchical structural relations of conventional product structures is feasible to a limited extent only.

An integrated product and process documentation must allow to manage experience and knowledge. The information must be available along the entire product generation process at any time. Existing hierarchical structural relations must be supplemented in a way to document explicitly existing knowledge in the product structure and make it available across projects.

For the generation of workplans, results and experience of earlier projects shall be available to be used for an (automated) deriving of workplans. To avoid errors and react towards problems early, methods for a product design which considers assembling aspects are required (cf. 2.3.4; 2.4.8; 2.4.9; 2.5.5; 2.5.6).

3.1.6 Realization of plausibility algorithms

A product structure defined by product documentation serves as link between the (numerous) digital views and the (single) physical representation of an individual customer vehicle.

Plausibility algorithms shall allow for the possibility to generate process specific views (cf. Fig. 3.1).

To increase process speed and improve both data up-to-dateness and quality, a procedure is required which allows for a computer aided documenting of assemblies. An IT solution is demanded which supports an automated workplan generation for the assembling sector. Based on part and procedure descriptions, the computer shall generate a workplan automatically. These workplans shall be available to downstream process functions and systems (cf. 1.1.2; 2.5.5; 2.5.6).

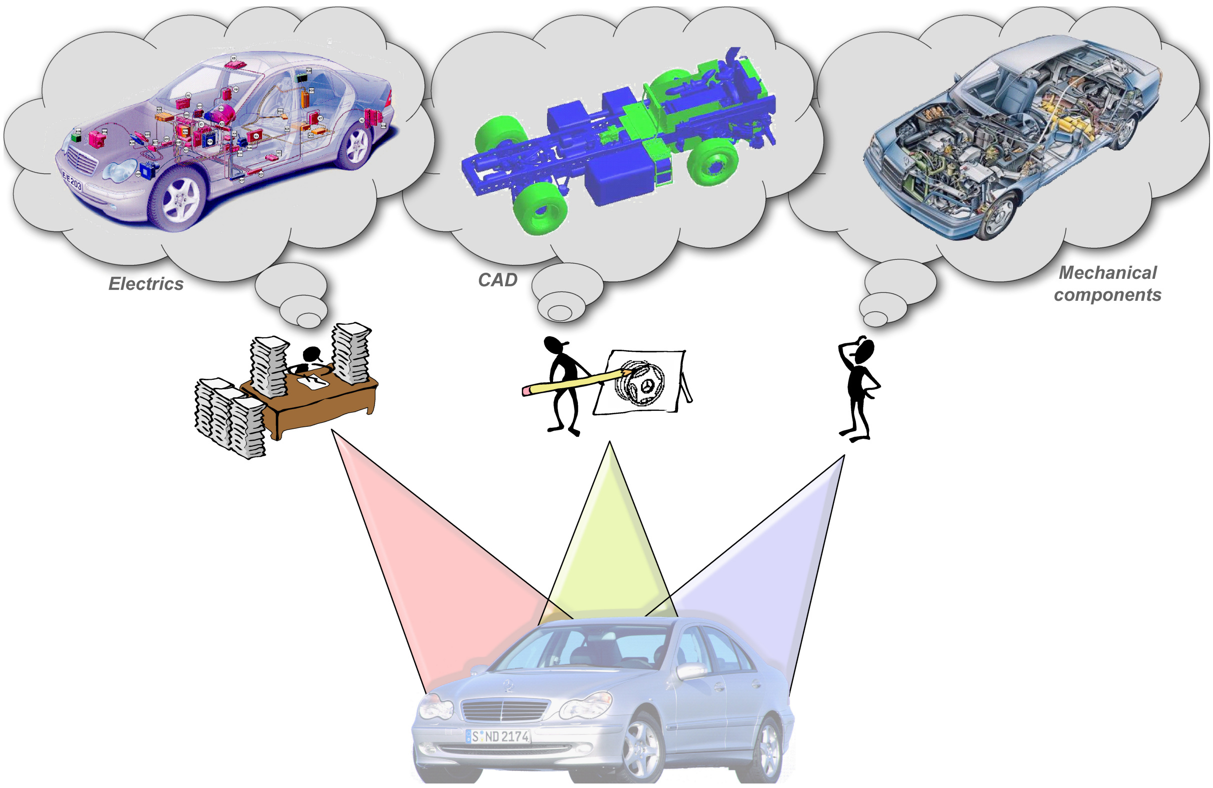


Fig. 3.1: Generating views based on plausibility algorithms

3.1.7 Reusing parts and components

With the increasing, global competitive situation and the associated cost pressure with a simultaneous demand for shorter product engineering times and increasing quality demands, it is a strategic aim of manufacturing companies to reuse parts and components across the entire product portfolio to obtain scaling effects. An integrated product and process documentation has to support this strategic aim of reusing parts and components (cf. 2.3.4; 2.4.2).

3.1.8 Providing a product schema across model ranges

Today, each product group (model range e.g.) has a separate product structure. This results in the disadvantage that parts which are used in several vehicle model ranges also have to be documented repeatedly, i.e. in a redundant way. Besides, it is very difficult to provide experience and knowledge obtained from major components or complex assemblies already in operation for different model ranges. The introduction and application of a product schema (cf. Fig. 3.2) shall meet the strategic target to make the existing and proven (technological) concepts available for both parts and components across all product groups.

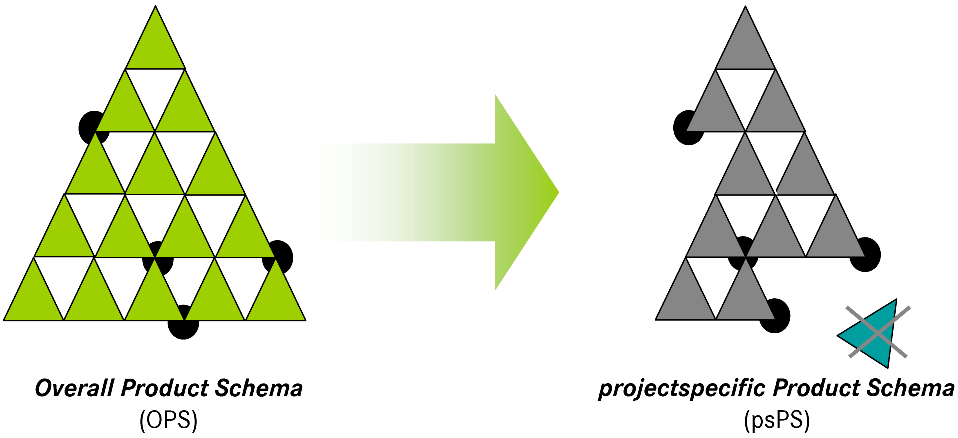


Fig. 3.2: Product schema

The product schema represents the reference structure for all derived project specific product structures of a manufacturing company. Amongst others, it contains the substantial components of the parts to be realized (cf. 2.3.4; 2.4.9).

3.2 Requirements as seen from process management

This section describes the requirements placed on an integrated product and process documentation as seen from process management. The product documentation method serves as the basis for process design and optimization.

3.2.1 Process management and process control

International networks, consisting of OEMs, system and part suppliers, design partners and sales partners have to be organized and coordinated globally. The method of *Technical Documentation* has to provide the basis for an instrument of process control inside and across companies.

A future oriented documentation method must not be limited to a static writing down of the manufacturing process, but must work as an instrument to support a dynamic product and process management. An improved integration of parts list information and the process data of production planning must improve communications between manufacturer and customer.

For a directed variant or complexity management, the structural relations of conventional documentation methods have to be expanded. The conventional methods allow for a mapping of effects on processes in particular caused by additional parts variants only to a limited extent (cf. 1.1; 1.1.1; 1.1.2; 1.1.3; 1.1.4; 2.2.8; 2.4.8; 2.4.9).

3.2.2 Process design and process integration

Designing flexible and continuous business processes is directly dependent on the method of product structuring and documentation. There must be the possibility to optimize the entire product lifecycle with regards to costs, period and quality. The method of an integrated product and process documentation shall provide a basis for a maximum integration of the two core processes of product generation and customer order processing. Based on a logical data model, avoiding redundant data, interfaces and friction losses are at focus (cf. 1.1; 1.1.2; 2.2.8; 2.4.9).

3.2.3 Seamless reuse of engineering data

The product structure is considered to be a central means of product documentation and serves as backbone along the entire product lifecycle. Typically, the product structure is created during product engineering. Other areas, like for example sales and marketing, work planning, calculation, procurement, production and maintenance use the data parallelly or downstream. The degree to which a seamless transition from digital product engineering to the physical world of vehicle production is supported is regarded as criterion for an ideal product structuring method. What is at issue is to provide a method which supports the continuity of processes and flexibility of organizational structures to the maximum. The method shall allow to provide the information generated in product engineering to the subsequent process functions without losses.

For a corporate wide continuous product generation process, there must be no splitting into design and manufacturing or assembling parts lists. It must be possible to derive the different parts lists computer aided from a joint database. Measures for an assembling favorable product design have to be considered at a very early stage in the design process. Parts and assemblies shall not be bundled arbitrarily, but shall be sensible from a technical point of view. Besides, the relation to manufacturing and assembling must be visible.

The interface between design and work planning shall not be limited to the exchange of CAD models, drawings and parts lists. Work planning must adequately support the central position between design and manufacturing. Logical correlations to recognize elements of manufacturing technology and the sequence of assembling steps shall be allowed for (cf. 1.1.2; 2.4.8; 2.4.4; 2.4.9; 2.5; 2.5.6).

3.2.4 Improvement of start-up management

Today's methods and IT implementations support the central position of the workplan between design and manufacturing only to a limited degree. To improve the start-up management, the interface between work planning and production planning has to be optimized. The integrated product and process documentation must actively support the optimization of the start-up phase (starting from the first customized vehicle to reaching peak production, cf. Fig. 3.3) (cf. 2.4.8; 2.5.6).

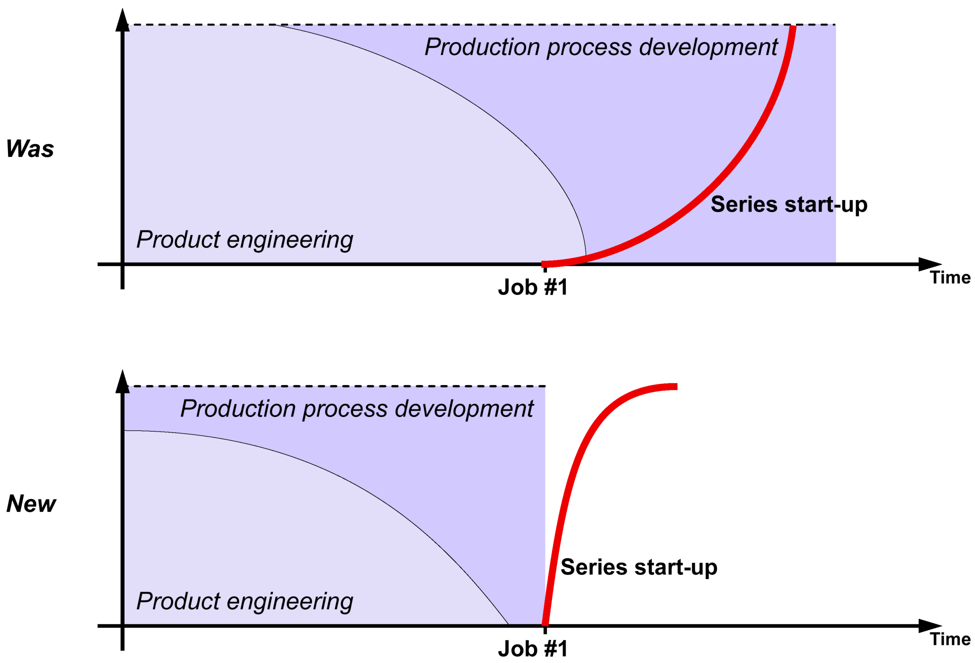


Fig. 3.3: Start-up management

3.2.5 Mapping the manufacturing or assembling sequence

To avoid errors not being recognized today during product engineering, work planning or parts manufacturing in future or to address such problems early, product designing methods living up to assembling requirements shall be provided. The integrated product and process documentation must support auxiliary means, like the preference graph and the manufacturing sequence which shows the operation sequence as single operations in a graphical form (cf. 2.4.4; 2.4.9; 2.5.2; 2.5.6).

3.2.6 Integrated change management

To counteract the strongly increased number of changes during the start-up phase (cf. Fig. 3.4), the integrated product and process documentation must support the central position of work planning between engineering/design and manufacturing. The identification of all parts and assemblies affected by a change shall be realized in an automated way.

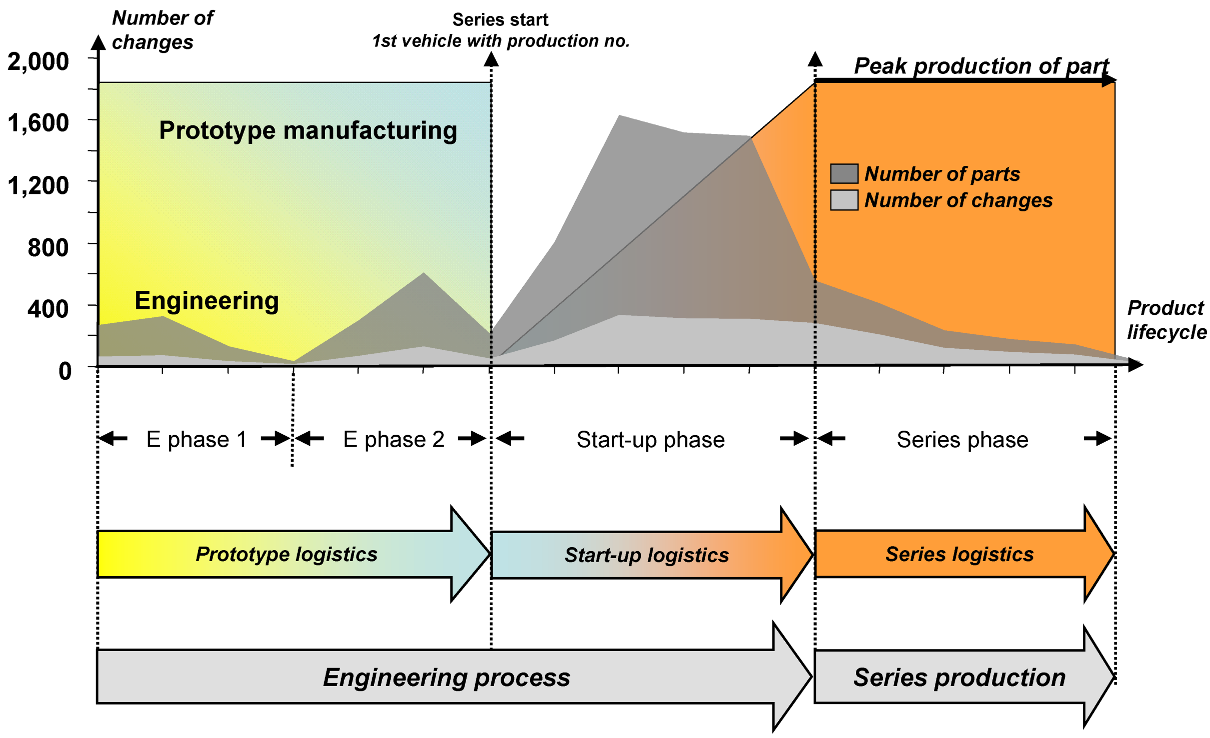


Fig. 3.4: Occurring changes during the product lifecycle

Not only shall the effects on the parts be transparent and rateable, but also the effects on the process side and the consequences for the manufacturing process. The identification or controlling of objects which are affected technically from a modification shall result in an improved process safety. Consistency checks shall be applied to additionally support the change management. The envisaged results are data which are more up-to-date with a better quality and an increased process safety (cf. 2.4.8; 2.4.9; 2.5.6).

3.2.7 Providing process specific information

The expectations placed on integrated product and process documentation include to support the provision of process specific information.

Parts and assemblies shall not be bundled arbitrarily, but shall be sensible from a technical point of view. Besides, the relation to manufacturing and assembling must be visible. Important work papers, like schedule card, wage and material slip, can be derived from the workplan. An integration of parts list information into the process data of production planning is necessary, for example, to adequately support feasibility restrictions on the production side for a communication between manufacturer and customer (cf. 2.2.8; 2.4.4; 2.4.9; 2.5.1;2.5.2; 2.5.3; 2.5.5; 2.5.6).

3.3 Requirements on IT solutions

This section describes the nonfunctional requirements placed on an integrated product and process documentation. To ensure the technical implementation of a software system, fundamental technical aspects have to be considered very early when developing a new method. This section will describe the requirements which are important when developing a prototype implementation:

- software ergonomics, usability, visualization
- provision, robustness and stability, efficiency
- migration capability, integratability, data consistency
- serviceability, expandability, scalability

The following requirements will not be detailed further:

- response times and data throughput of the system
- availability and reliability
- network topologies, used protocols
- security management

3.3.1 Software ergonomics, usability, visualization

Due to missing graphical tools, structural data were often represented as lists in the past. With the development of graphical user interfaces and the introduction of browsers, list representations should be turned down to the benefit of graphical representations of product and parts list structures.

The use of the application (the prototype solution in particular) should be intuitive and easy to the greatest possible extent. For complex *workflows*, users shall be prompted by the system. Possible "wrong entries" or operational errors by the system shall be avoided to live up to *usability engineering* (= software ergonomics).

A clearly structured GUI shall present information clearly laid out where the essential elements of the documentation method are easy to recognize (cf. 2.3.5).

3.3.2 Provision, robustness and stability, efficiency

For the use of the software solution, it is desirable that the software can be run on customary operating systems. A fast and easy installation of the software is assumed as is efficient working.

3.3.3 Migration capability, integratability, data consistency

A variety of system interfaces consequently results in redundant, inconsistent data, outdated data, and poor data quality. To avoid a heterogeneous system world, interfaces shall be avoided wherever possible, and standards shall be used (cf. 1.1.2; 1.1.3).

3.3.4 Serviceability, expandability, scalability

Modern IT systems have to be easily expandable to react flexibly towards new requirements. Software systems to be developed newly have to offer a high degree of serviceability to support a long system life and keep the maintenance effort low. Expandability in particular decisively depends on the used software architecture. To live up to modern software engineering, this architecture should be structured in layers and use defined interfaces and software draft patterns.

3.4 Summary

This section gives an overview of the requirements placed on the interconnections documentation described in sections 3.1 to 3.3. Table 3.1 shows the reference to the conceptual implementation in chapter 4 and the design and prototype implementation of the interconnections documentation in chapter 5 and 6.

Table 3.1: Overview of requirements placed on the interconnections documentation to be considered in the conception and prototype implementation

Requirements	Concept	Prototype
Requirements as seen from variant management		
Integration of product, process and resource information (cf. 1.1.1; 1.1.2; 1.1.3; 2.2.4; 2.2.8; 2.4.8; 2.4.9; 2.5.2; 2.5.3; 2.5.6)	4.1.1; 4.1.3; 4.1.4; 4.2.1; 4.2.3; 4.3.2; 4.3.4; 4.3.5; 4.5.1; 4.5.2; 4.5.3; 4.5.4; 4.7.2; 4.8.3; 4.9.1; 4.9.3	6.1.2; 6.1.4; 6.2.3; 6.2.4; 6.1.8

Requirements	Concept	Prototype
Integration of product engineering and production planning (cf. 1.1.1; 1.1.3; 2.2.8; 2.4.4; 2.4.8; 2.4.9; 2.5, 2.5.1; 2.5.2; 2.5.3; 2.5.5)	4.1.1; 4.1.2; 4.1.3; 4.2.3; 4.3.1; 4.3.2; 4.3.4; 4.3.5; 4.4.2; 4.5.1; 4.5.2; 4.5.3; 4.5.5; 4.8.1; 4.8.2; 4.8.3; 4.9.2; 4.9.3	6.1.2; 6.1.4; 6.2.3; 6.2.5; 6.2.6; 6.2.8
Flexible and easily expandable data model (cf. 2.4.8; 2.4.9)	4.1.1; 4.3.2; 4.8.2; 4.8.3; 4.9.3;	6.1.2; 6.2.3; 6.2.8
Unambiguous and redundant free mapping of information (cf. 1.1.1; 1.1.3; 1.1.2; 2.2.8; 2.4.2; 2.4.8; 2.4.9)	4.1.1; 4.1.3; 4.1.4; 4.2.1; 4.2.2; 4.3.2; 4.3.4; 4.3.5; 4.5.5; 4.8.1; 4.8.2; 4.8.3; 4.9.2	6.1.2; 6.1.4; 6.2.3; 6.2.4; 6.2.6;
Documenting experience and knowledge (cf. 2.3.4; 2.4.8; 2.4.9; 2.5.5; 2.5.6)	4.1.1; 4.2.1; 4.3.2; 4.8.3; 4.9.2	6.1.2; 6.1.4; 6.2.3; 6.2.5
Realizing plausibility algorithms (cf. 1.1.2; 2.5.5; 2.5.6)	4.1.4; 4.5.5; 4.8.1;4.8.2; 4.8.3	6.1.3; 6.2.6
Reusing parts and components (cf. 2.3.4; 2.4.2)	4.1.2; 4.2.1; 4.2.2	6.1.1; 6.2.2
Providing a product schema across model ranges (cf. 2.3.4; 2.4.9)	4.2.1; 4.2.2; 4.2.3; 4.3.4	6.2.2
Requirements as seen from process management		
Process management and process control (cf. 1.1; 1.1.1; 1.1.2; 1.1.3; 1.1.4; 2.2.8; 2.4.8; 2.4.9)	4.1.1; 4.1.3; 4.1.4; 4.2.1; 4.2.2; 4.2.3; 4.3.4; 4.3.5; 4.4.2; 4.5.1; 4.5.2; 4.5.3; 4.5.4; 4.5.5; 4.8.1; 4.8.2; 4.8.3, 4.9.1; 4.9.3	6.1.2; 6.1.3; 6.1.4; 6.2.2; 6.2.3; 6.2.4; 6.2.5; 6.2.6; 6.2.8
Process design and process integration (cf. 1.1; 1.1.2; 2.2.8; 2.4.9)	4.1.1; 4.1.3; 4.3.4; 4.3.5; 4.4.3; 4.5.1; 4.6.2; 4.6.3; 4.6.4; 4.7.2; 4.8.1; 4.8.3; 4.9.1; 4.9.2; 4.9.3	6.1.2; 6.1.3; 6.1.4; 6.2.2; 6.2.3; 6.2.4; 6.2.8
Seamless reuse of engineering data (cf. 1.1.2; 2.4.8; 2.4.4; 2.4.9; 2.5; 2.5.6)	4.1.1; 4.1.3; 4.2.2; 4.2.3; 4.3.4; 4.3.5; 4.5.1; 4.5.2; 4.5.3; 4.5.4	6.1.2; 6.1.4; 6.2.2; 6.2.3; 6.2.4; 6.2.8
Improvement of start-up management (cf. 2.4.8; 2.5.6)	4.1.1; 4.1.3; 4.3.2; 4.3.4; 4.3.5; 4.9.1	6.1.2; 6.1.4; 6.2.3; 6.2.4;
Mapping the manufacturing or assembling sequence (cf. 2.4.4; 2.4.9; 2.5.2; 2.5.6)	4.1.1; 4.3.4; 4.3.5; 4.5.1; 4.5.2; 4.5.3; 4.9.3;	6.1.2; 6.2.8;
Integrated change management	4.1.1; 4.1.3; 4.3.4;	6.1.2; 6.2.4;

Requirements	Concept	Prototype
(cf. 2.4.8; 2.4.9; 2.5.6)	4.3.5; 4.9.1	6.2.8
Providing process specific information (cf. 2.2.8; 2.4.4; 2.4.9; 2.5.1;2.5.2; 2.5.3; 2.5.5; 2.5.6)	4.1.1; 4.1.3; 4.2.3; 4.3.2; 4.3.4; 4.3.5; 4.5.3; 4.5.5; 4.8.1; 4.8.2;	6.1.2; 6.1.4; 6.2.3; 6.2.4; 6.2.6; 6.2.8;
Requirements on IT solutions		
Software ergonomics, usability, visualization (cf. 2.3.5)	4.2.1; 4.3.3; 4.9.2; 4.9.3	6.1.1; 6.1.2; 6.1.4; 6.2.2; 6.2.3; 6.2.4; 6.2.5; 6.2.6; 6.2.8
Provision, robustness and stability, efficiency		5.2.1; 5.2.2; 5.2.3
Migration capability, integratability, data consistency (cf. 1.1.2, 1.1.3)		5.1.1; 5.1.2; 5.3.1; 5.3.5
Serviceability, expandability, scalability		5.2.1; 5.3.1; 5.3.2; 5.3.3; 5.3.4; 5.3.5

Part III

CONCEPTUAL LAYOUT

4 Concept of the interconnections documentation

This part of the study deals with the concept of the integrated product and process documentation based on connections (in short: interconnections documentation). Section 4.1 presents the method of the interconnections documentation in an overview. Afterwards, the key concepts of the method will be described in detail in the sections 4.2 to 4.8. An example of the integrated use of parts and interconnections will be given in section 4.9. The migration concept presented in section 4.10 and a summary of the interconnections documentation concept in section 4.11 will conclude this part of the study.

4.1 Defining and deriving the interconnections documentation

By defining and deriving the interconnections documentation, this section presents the key concepts of the interconnections documentation in an overview. At first, section 4.1.1 introduces the underlying understanding for the interconnections documentation that a product is created by "connecting" parts. This is followed by a description of the interconnection based top-down procedure of product structuring described in section 4.1.2. Starting from the variant and complexity management based on parts and interconnections (cf. section 4.1.3), section 4.1.4 explains the process specific view generation. A summary in section 4.1.5 concludes this part of the study.

4.1.1 Product documentation as a web of parts and interconnections

The interconnections documentation follows the trivial assumption that a product is made up by "connecting" parts. Hence, a product in the sense of the interconnections documentation is more than the sum of its parts. It consists of parts and the connections between these parts. As shown in Fig. 4.1, products can be described as webs. Apart from the parts[1], the interconnections[2] are a substantial means "on an equal footing" to describe a product.

1 Represented with circles in graphics.

2 Represented with edges in graphics.

As independent objects and to obtain a complete product description, the interconnections can hold process specific information (cf. Fig. 4.1).

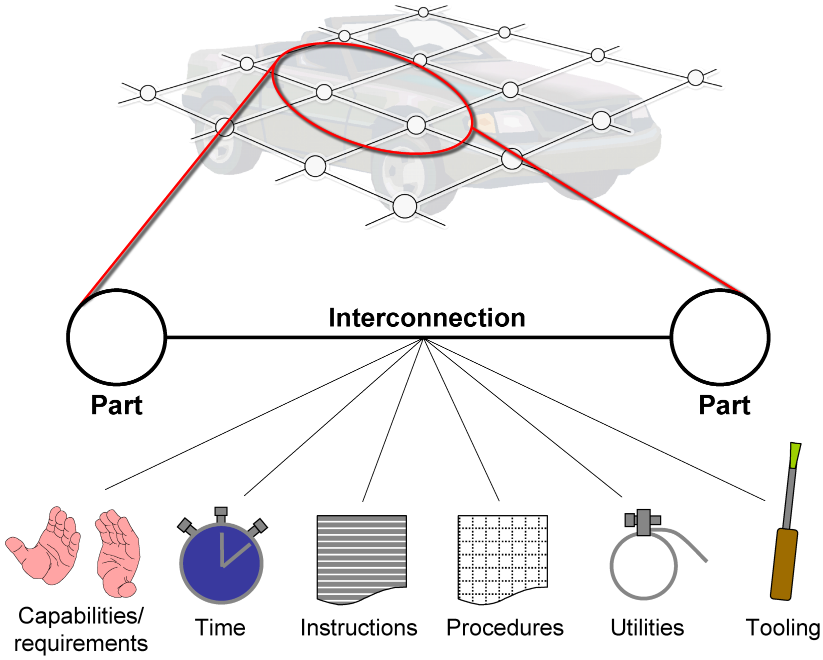


Fig. 4.1: Representing a product as web of parts and interconnections

4.1.2 Top-down to parts and interconnections

The product as a web of parts and interconnections is generated step by step in the product generation process by a company specific splitting (= product structuring) with the top-down procedure. Starting from the project description, each separating step adds new details to the web. As the structural complexity[1] of the web increases with every detailing step, the simplicity[2] of the individual web nodes (i.e. parts and components) increases also. This process is repeated down to the defined corporate design level (= lowest level of web structure).

The conception and definition of the product schema for the entire product portfolio of a company as described in section 4.2 simplifies and standardizes this process of splitting up.

The schema called "overall product schema" (OPS) forms the reference structure for all derived project specific product structures (psPS) of a manufacturing company. Amongst others, it contains the substantial constituents of the products to be built. Besides, the schema serves as navigation means to access objects and data of the product documentation and identify them fast and easily. This provides for a greater transparency across the respective product classes[3] and allows for a comparison of corporate products.

1 Complexity means the increasing degree of cross-linking by additional web edges.

2 Simplicity means the degree of features and difficulty of a product component represented by a node.

3 Model ranges, for example.

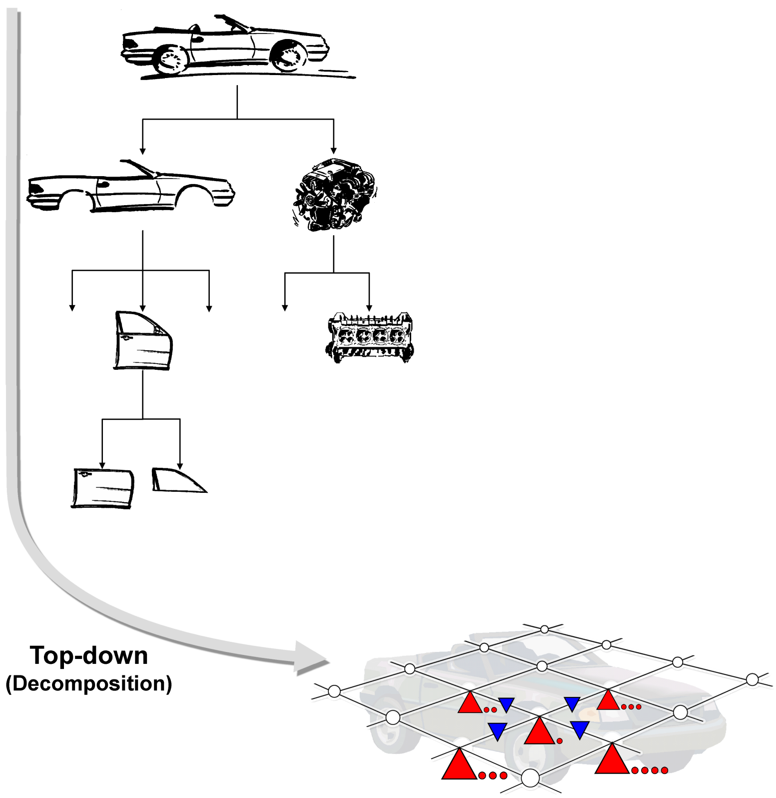


Fig. 4.2: Hierarchical product representation as web

Parts and components including their structural associativity are transparent across all products and available for reuse in different configurations. Hence, the strategic target of manufacturing companies to achieve scaling effects is met.

4.1.3 Variant diversity and complexity management

Variant diversity of products and the resulting complexity in the business processes is created on the lowest level of the product description by adding, omitting or replacing parts at defined positions. These positions represent geometric locations (where-used locations) or design envelopes within a product. In the context of the interconnections documentation, they are referred to as part positions (PP).

Part positions bundle 1 to n part position variants (PPV) to manage parts which are used as alternatives. The part position variants store all activity data regarding design and procurement (e.g. geometry, suppliers, parts costs, inventory data, ...). One part position variant for each part position represents the basic version variant ("series" status) of a product.

Short code rules with merely positive expressions control when which part position variant of a part position is used, i.e. which part is to be used for the respective vehicle or product variant.

Manufacturing activities represent the largest portion of all activities since after designing and procuring the parts, all activities of the areas involved in the manufacturing process entwine around connecting them together (manufacturing planning, tooling design and construction, work and capacity planning, assembling process, ...). As for the representation of part positions and part position variants, the interconnections documentation offers another class of key terms, namely interconnection positions (IP) and interconnection position variants (IPV).

Interconnection positions and the interconnection position variants they might contain define relations (physical and/or logical relations) between the part positions and part position variants. Hence, interconnection positions and their interconnection position variants map all manufacturing activities in a process appropriate way to provide the basis for a horizontal integration of product and process information[1].

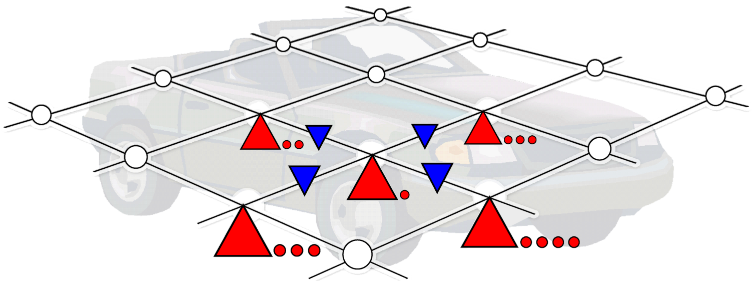


Fig. 4.3: Representation of a basic web
considering part positions and interconnection positions

As for the part position variants, short code rules with merely positive expressions are used to control when which interconnection position variant is relevant for a customer order (cf. section 4.3.3).

The introduction of part positions and part position variants together with interconnection positions and interconnection position variants respectively provides for the basis of describing a product as a web of parts and interconnections. Thus, a horizontal integration of the product and process documentation is realized.

The product web based on the smallest design elements (SDE) is called SDE web or basic web (cf. section 4.3.4). The detailed methodological procedure to build and update this product web is described in section 4.3.5.

4.1.4 Process specific views and bottom-up generation

Starting from the basic web, interconnection positions and the interconnection position variants contained there can be used to map functional views along the core processes of "product generation" and "customer order processing" from the viewpoint of processes.

Bottom-up algorithms can be used, for example, for a (completely) automated generation of vehicle specific workplans and to map the manufacturing status of a vehicle at any point in

1 This lives up to the "second level completeness" called for in section 1.1.1.

time (cf. Fig. 4.4). The bottom-up method and the process specific view generation are explained in section 4.8.

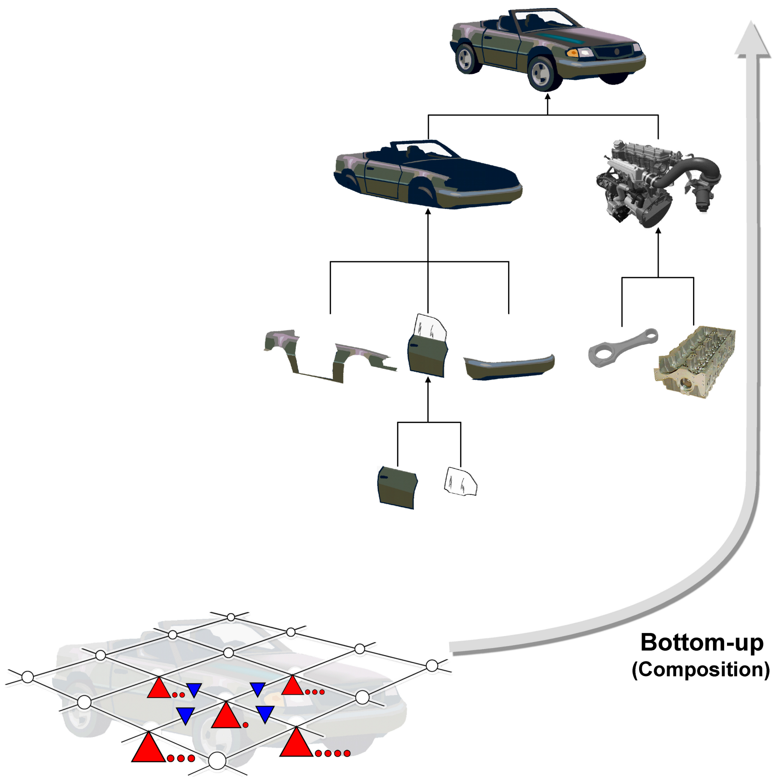


Fig. 4.4: Principle representation of bottom-up generation

4.1.5 Summary and further procedure

The essential concepts of the interconnection based product and process documentation include:

- definition of the overall product schema (OPS) including the method to derive the project specific product structure (psPS)
- building and maintaining the product web based on parts and interconnections (= SDE or basic web)
- process specific view generation via bottom-up generation

Now, the rough concept for an integrated product and process data model (IPDM) is described fulfilling the requirement both for a continuous product generation process starting from product engineering to product creation (cf. Fig. 4.5) and the integration of the core processes of "product generation" and "customer order processing".

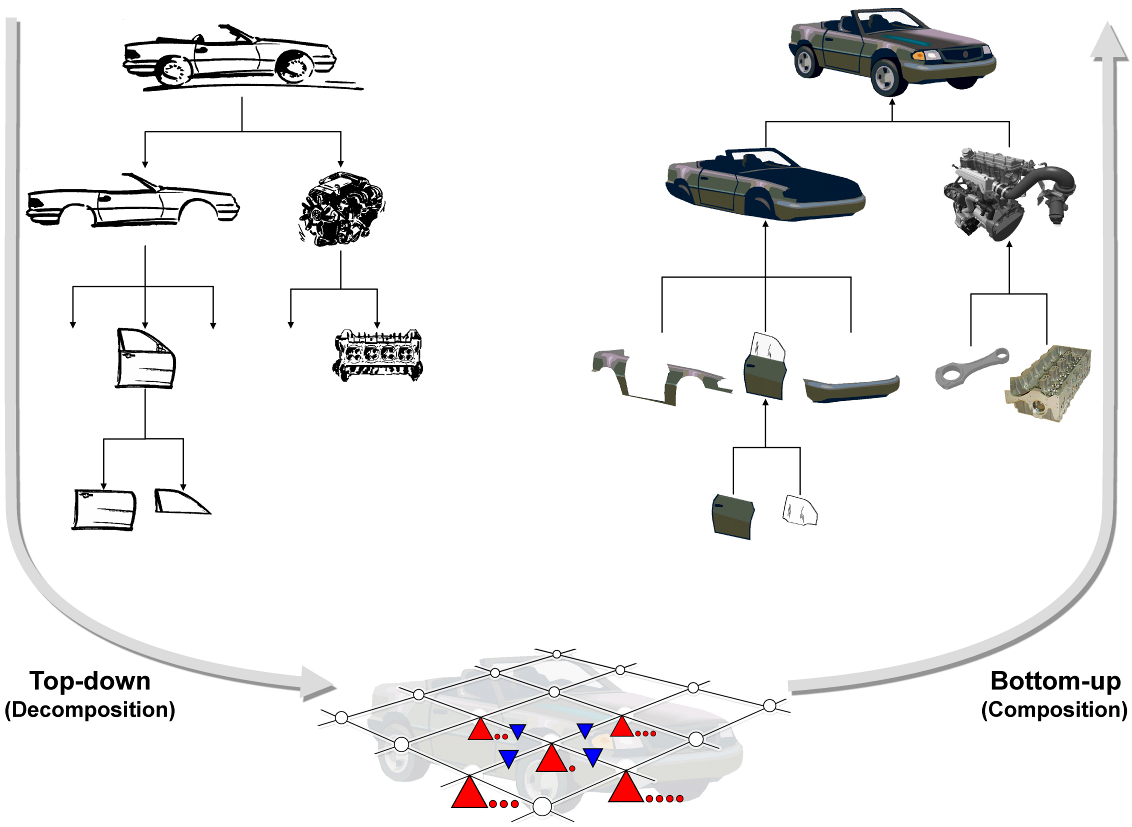


Fig. 4.5: Principle representation of the interconnections documentation

For a further detailing, section 4.2 gives a close description of the product schema followed by the methodological procedure of building and maintaining the basic web described in section 4.3. With the results of this section, an instrument for an active variance and complexity management based on parts and interconnections is available.

The methodological concept of decomposition is used to describe a procedure to refine the product web step by step in section 4.4.

To map the manufacturing flow of a vehicle via the sequence of the interconnection positions, the concept of the preference graph and manufacturing sequence is introduced in section 4.5. Starting from sequencing, this section also describes the assembly generation.

Section 4.6 addresses the mutual effects of design and manufacturing depth and their consequences for documentation, while section 4.7 explains the concept of technical feasibility in the context of the interconnections documentation.

Section 4.8 details the process specific view generation via bottom-up generation. The conceptual approach of the change management based on the interconnections documentation is described in section 4.9.

While section 4.9 introduces the integrated use of parts and interconnections within IPDM, section 4.10 addresses a few aspects of a step-by-step migration to close with.

4.2 OPS and psPS

In the following, the concept of OPS (overall product schema) and the procedure how to derive the psPS (project specific product structure) are described. In this context, reference is made to the product schema of the commercial vehicles division at Daimler AG which was developed in the course of this study. In section 4.2.1, the general structure and the key terms of the OPS are defined, followed by statements on the use of the OPS in the product generation process and how to derive the psPS from the OPS given in section 4.2.2. Sections 4.2.3 and 4.2.4 deal with the management of process specific information in the OPS and the organizational embedding of the product schema.

4.2.1 Definition and structure of the OPS

An OPS is a vehicle neutral structure which is valid for all product classes to manage the entity of all system components in a company.

Functional criteria seem to be most appropriate to structure the schema[1]. In an OPS structured by functions, system components represent both vehicle functions and concrete assemblies or single parts of these functions.

In the OPS, the system components are represented by different position types. There are "conventional" relations of the type "part of" or "used by" between the positions. In this way, an OPS can be represented as multi level, hierarchical structure.

The OPS defined for the Truck Group[2] at Daimler AG is a five level product schema based on functional-geometric criteria (cf. Fig. 4.6). The entry position on the top level (level 0) of the schema is called "Vehicle". The positions of the next lower level (level 1) represent the vehicle systems "Mechatronics", "Chassis", "Cab", and "Powertrain". Positions on this level are called "navigation systems". The four vehicle systems are split up to a total of 62 vehicle functions on the next structural level (level 2). Positions on this level are called "navigation groups" which, in turn, are split into 905 "functional modules" (level 3) and 1,405 "placeholders" (level 4).

1 Changes in vehicle functions occur significantly less often than changes in organization.

2 Mercedes-Benz Trucks, MFTBC, Freightliner (including Thomas Built Buses), Evobus and Power Systems (including Axle Alliance Corporation and Detroit Diesel Corporation).

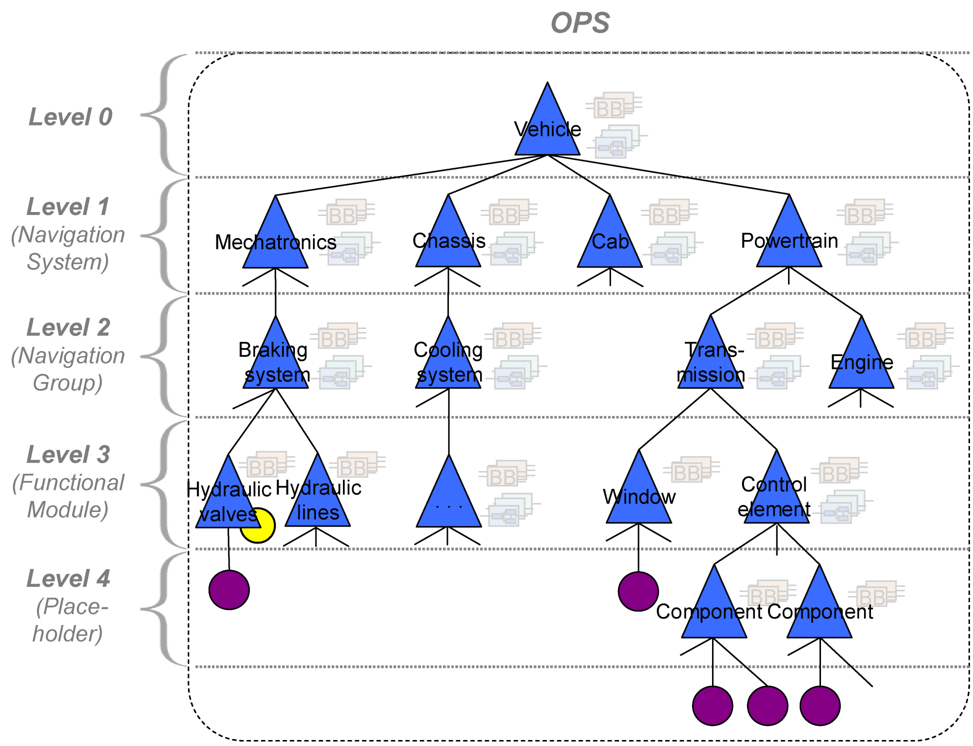


Fig. 4.6: Overall Product Schema

Irrespective of the corresponding structural level, each position administrates one or several specifications. These objects called "blackbox specifications" (BB) define the functional requirements and the interface(s) of the position with effects to the outside. Each "blackbox specification", in turn, has a defined number of corresponding "implementation templates". These templates describe the internal functions, components or parts (positions of a lower structural level) and their structural associativity. Each "implementation template" represents a web of positions and interconnections which is specific for the respective structural level.

To live up to the commonality idea of reusing preferred parts or shared parts and components, it is possible to manage concrete physical parts and assemblies at the positions of the OPS. As shown in Fig. 4.6, schema elements which manage preferred parts or shared parts and components are identified appropriately.

4.2.2 Deriving the psPS from the OPS

Based on the OPS, a vehicle is developed in a top-down way with a step-by-step deriving of a psPS. Starting from the vehicle systems of the OPS (see above), this process of iterative deriving creates an increasingly more detailed product model with the advance in engineering progress as web of positions and interconnections. This detailing process which is described in the following is repeated down to the level of the basic web (cf. section 4.3). Fig. 4.7 shows an example.

For each new product project (new vehicle model range, for example), the project manager in charge derives a psPS from the OPS. At first, the top position node of the OPS is selected, and a corresponding reference node, the top level data object (TLDO) is instantiated for the project specific product structure (cf. step 1 in Fig. 4.7). The TLDO manages the project outlines which describe the market requirements and the market potential of the product project. The vehicle-related characteristics and features as well as the generally valid customer requirements of the outline specifications serve as basis for the next detailing step which describes the framework specifications. The framework specifications document the requirements which are identified with regards to market and customer desires, environmental parameters, and corporate strategies. The technical vehicle concepts are specified as blackbox specifications which are also stored at the TLDO of the psPS. Allowing for make-or-buy decisions and commonality requirements, the blackbox specifications of the framework specifications are used to select an appropriate implementation template. Instantiating this implementation template provides a first psPS as product web (cf. step 2 in Fig. 4.7). On this level, the product web consists of the vehicle systems and their associativity. The instantiated system components of the psPS reference the blackbox specifications of the navigation groups in the OPS. On the level of the system components, the design managers and component representatives are responsible for selecting and describing the blackbox specifications as well as instantiating the corresponding implementation templates. On this level, the implementation templates describe the vehicle functions and their associativity (cf. step 3 in Fig. 4.7).

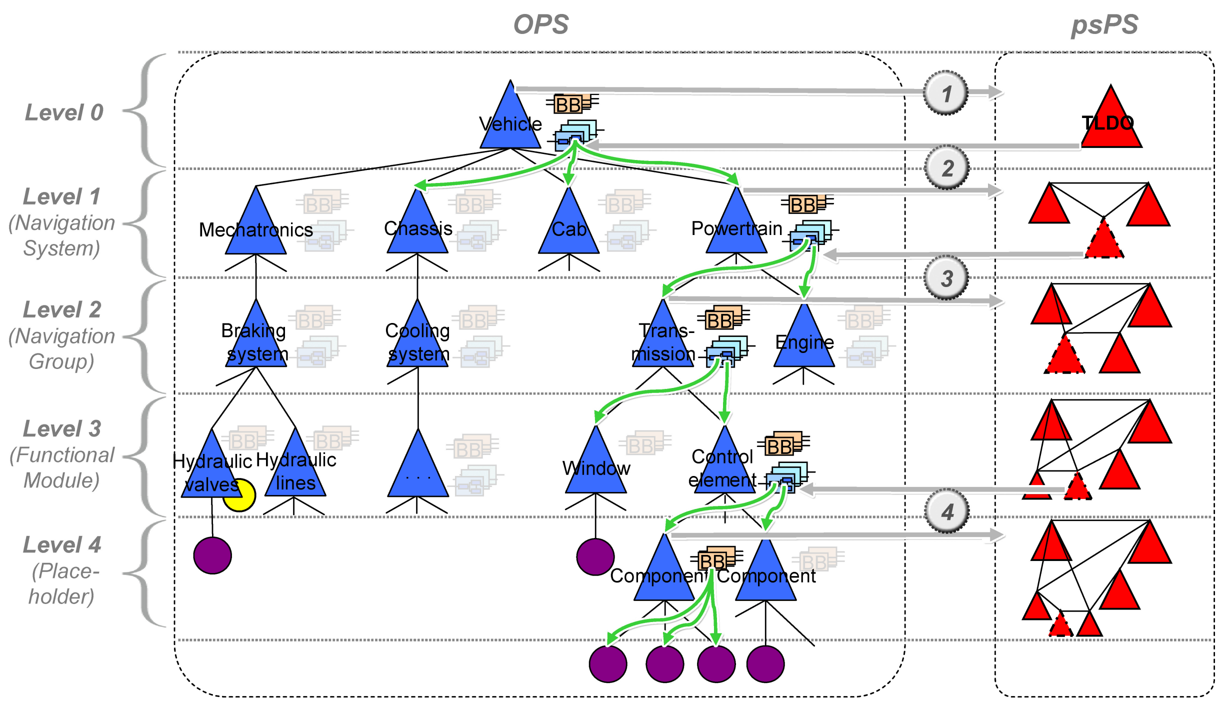


Fig. 4.7: OPS and psPS derived step by step

In this way, the project team refines the product structure down to the level of functional modules and placeholders (cf. step 4 in Fig. 4.7). In a first approximation, a project specific product structure as basic web consisting of parts and interconnection positions is available.

If the product schema contains concrete parts or components, these are parts to be reused in the sense of preferred parts or shared parts and components. Such parts and components shall be adopted from the OPS to the specific product structure.

4.2.3 Process specific information in the OPS and psPS

Selecting the blackbox specifications and the implementation templates does not only determine decisions on product components and their technological concepts but also determines or restricts manufacturing methods. Process specific information can already be stored in the OPS to allow for an early use and consideration when instantiating a psPS.

The possibility to store manufacturing information in the OPS apart from product specific data and provide them corporate (or project) wide can also be transferred to other functional areas. Supplier and cost information from previous products, for example, can be managed in the OPS and made available in the psPS.

4.2.4 Organizational embedding of the OPS

An independent organization is responsible of collecting and coordinating requirements on engineering which shall be provided to the engineering sectors in charge. The designers or design groups have to create the respective templates of a schema element.

Placeholders on the bottom level of the schema or the specific product structure might have to be refined further. In this case, blackbox specifications are defined, and the placeholders adopted from the schema are detailed further by splitting them up in partial webs (consisting of positions and their interconnections). The partial webs developed for existing placeholders this way can, in turn, be stored as templates in the OPS to be available for future product projects.

4.3 Product configuration

As basis for an integrated product and process data model (IPDM), this section will describe how the smallest design elements (SDE) and their interconnections are documented as product web on the lowest level of the product structure. Sections 4.3.1 to 4.3.4 give a description and definition of the key terms and structuring elements of the interconnections documentation. The subsequent section 4.3.5 describes in detail how the SDE web is built and maintained.

4.3.1 Part positions and part position variants

In the sense of the interconnections documentation, a product (vehicle model range for example) is split completely and without clashes into part positions and interconnection positions.

A part position (PP) represents a geometrical location (where-used location) or design envelope within a product. It administrates all parts which can be used at a defined position with the same function as alternatives (cf. section 2.4.7). If a part has different where-used locations, this is mapped by representing the part in several part positions. Each part position always has at least one part position variant (PPV)[1]. The parts information associated with this part position variant represents the basic version of a product group. For each part position, additional part position variants can be administrated as alternatives to be used. In principle, the part position variants store all activity data regarding design and procurement (e.g. geometry, suppliers, part costs, inventory data, ...).

Considering rule elements (short code rules, cf. section 4.3.3) and dates, the conditions are defined when the respective part position variant, i.e. part, is used in the product.

According to the above statements, a part position can be defined as follows:

> A part position (PP) corresponds to a defined spot in the product and bundles 1...n alternative part position variants (PPV) for one part position.
>
> Part positions are used to address a geometrical location within a product thus describing a location the part can be mounted to.
>
> *Part positions are represented with the graphical element of a (red) triangle.*

Considering what has been said so far, part position variants can be defined as follows:

> Part position variants (PPVs) are defined as mutually excluding alternatives within a part position (PP). One part position variant is assigned exactly to one part position. It is selected within a part position via a rule-based term.
>
> *Part position variants are represented with the graphical element of a (red) circle.*

Each part position of a manufacturing order addresses exactly one part position variant. The selection of the respective part position variant is based on a customer wish and is determined via the short code rule.

1 If no part at all is used at a specific part position on special conditions, this fact is documented via a part position variant "No part".

The key term[1] for selecting a part position variant is composed as follows:

- part position
- condition (short code rule[2])
- maturity level[3]
- part (incl. version, color)

The key term can be supplemented by additional elements in accordance with the company requirements.

4.3.2 Interconnection positions and interconnection position variants

In the sense of the interconnections documentation, an interconnection describes the correlation of two parts of a product and offers the option to manage attributes which are not managed unambiguously or without redundancy in conventional parts lists. Thus, interconnections map all manufacturing activities in a way which is in line with process requirements.

Interconnection positions (IP) describe a general relation (physical or logical, e.g.) between two part positions (cf. Fig. 4.8). They manage all (process specific) information of this relation and bundle the operations relevant for the interconnection. Hence, all process relevant information can be variant forming for the interconnection position apart from the operations and parts.

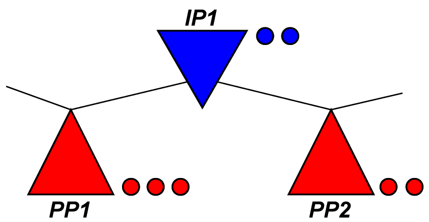


Fig. 4.8: Example of an interconnection representation allowing for part and interconnection positions

Interconnection position variants (IPV) describe the relation of two part position variants. Interconnection position variants can be used to describe different manufacturing processes or procedures for the same combination of two part position variants.

According to the characteristics described above, an interconnection position can be defined as follows:

1 It might be necessary to consider additional required key relevant attributes.

2 Including control codes.

3 The maturity level defines the quality level a position variant is documented with.

> An interconnection position (IP) is the central object of an interconnection and describes any relation (e.g. logical, physical) between two part positions (PP). It bundles 1...n interconnection position variants (IPV).
>
> *Interconnection positions are represented with the graphical element of a downward pointing (blue) triangle.*

Considering what has been said so far, interconnection position variants can be defined as follows:

> Interconnection position variants (IPVs) are defined as mutually excluding alternatives within an interconnection position (IP). An interconnection position variant is assigned exactly to one interconnection position, and is selected via a rule-based term within an interconnection position.
>
> *The maximum number of all theoretically possible variants of an interconnection is determined by combining the part position variants of the basic web (cf. section 4.3.4*) of the different positions of an interconnection.
>
> *Interconnection position variants are represented with the graphical element of a (blue) circle.*

The (parts) manufacturing process can be regarded as a special case for applying an interconnection. The relation between single part and raw material or rough part can also be represented via an interconnection position. The interconnections or their operations receive the attributes which are required to manufacture the single part. The single part is the result of the interconnection.

As for the key term of the part position variant, the key term for selecting an interconnection[1] is composed as follows:

- interconnection position
- condition (code rule[2])
- maturity level[3]
- interconnection

The key term can be supplemented by additional elements in accordance with the company requirements.

1 It might be necessary to consider additional required key relevant attributes.

2 Including control codes.

3 The maturity level defines the quality level a position variant is documented with.

The interconnection positions and interconnection position variants represent a second class of key terms which realizes the horizontal integration of product and process information in the IPDM[1]. Interconnection positions serve as integration object of the different product and process information providing the basis for a control instrument along the entire product generation process (cf. section 4.3.5).

As the basis for a complete mapping of all product variants of a product class[2], interconnections provide for the prerequisite both for assembly generation and the realization of process specific views (cf. section 4.8).

Each interconnection position of a customer order selects exactly one interconnection position variant. The respective interconnection position variant is determined via the short code rule. The following section will address the specific meaning of (short) code rules in this context.

4.3.3 Short code rule with positive expressions

The code rules used in the context of the interconnections documentation are short code rules with merely positive expressions.

The short code rules are created from the (individual) codes linked with the "+" operator[3]. They describe the conditions which have to be met so that the position variants are selected for a customer order. A decisive advantage for the documentation of the code rules results in combination with the BCT method[4]. The advantage lies in the shared viewing of the alternative parts of a position and a reduction of the code rules (cf. also [Ohl00]).

In the context of the interconnections documentation, this advantage is used for both the part and interconnection positions. The requirements for using short code rules shall be explained which were already described in section 2.4.7:

- Each position must consider the position variants for all combinations of codes that affect the respective position. This is absolutely mandatory since this is the only way to ensure that exactly one position variant per position will be selected for a customer order to be produced.
- Short code rules must not be considered isolated, but always in connection with the short code rules of the respective alternative parts.

For a customer order to be produced, the long code rule can be omitted, if the position variants of each position are evaluated in the sequence from the longest to the shortest code

1 This lives up to the "second level completeness" called for in section 1.1.1.

2 This lives up to the "first level completeness" called for in section 1.1.1.

3 This corresponds to the logical "and" of Boolean algebra.

4 Cf. section 2.4.7.

rule. The first code rule of a position that matches the customer order determines the part to be used or provides the corresponding interconnection information[1].

In literature, the practical use of code rules is widely rated rather critical. In the following, a few remarks on a pragmatic application (in the context of the short code rule) shall be formulated.

An example will be used where a position is affected by three codes (C1, C2, C3). Hence, eight position variants or code rules have to be considered at the respective position. Fig. 4.9 a) shows all short code rules to be considered for this example. Apart from the "empty" code rule (status "series"), the conditions C1, C2, C3, C1+C2, C1+C3, C2+C3 and C1+C2+C3 have to be documented[2].

a) Code

C1	C2	C3	Code rules
0	0	0	;
0	0	1	C3;
0	1	0	C2;
0	1	1	C2+C3;
1	0	0	C1;
1	0	1	C1+C3;
1	1	0	C1+C2;
1	1	1	C1+C2+C3;

b) Additional code / Additional code rules

C4	Code rules
0	
1	C4;
0	
1	C3+C4;
0	
1	C2+C4;
0	
1	C2+C3+C4;
0	
1	C1+C4;
0	
1	C1+C3+C4;
0	
1	C1+C2+C4;
0	
1	C1+C2+C3+C4;

Fig. 4.9: Code rules for code C1, C2, C3, and C4

The example shows the advantages of code rules documented in positive expressions:

- Short code rules with merely positive expressions can be understood and documented easily.
- Modifications which are caused, e.g., by a new code that affects the position can be integrated very easily. The additional code rules to be considered result from the already existing code rules with the new code linked with the AND operator. For the example above, the additional code rules C4, C1+C4, C2+C4, C3+C4, C1+C2+C4,

1 This procedure avoids the necessity of building long code rules (great effort/computing resources) by expanding the short code rule by the complementing remaining codes of the alternative position variants.

2 The negative shares of a code rule can be neglected without losing information.

C1+C3+C4, C2+C3+C4, and C1+C2+C3+C4 have to be considered for code C4 (cf. Fig. 4.9 b).

- The generation of the long code rule can be neglected when the code rules are interpreted from the longest to the shortest code rule when evaluating a position. Apart from clarity and transparency, this offers a decisive advantage for the user regarding a computer aided processing and evaluating of the code rules.

If the example which is restricted to three and four codes is expanded to any number of codes without loss of generality, the following shall be observed:

For a position which is affected by n codes, 2^n code rules have to be documented. Hence, 2^{n-1} additional code rules have to be managed for an additional code to be considered.

With regards to this immense number of code rule variants which seem to be uncontrollable manually in practice, it shall be mentioned that a large number of the code rules do not occur in practice[1]. These code rules resulting from the combination of mutually excluding codes[2] can already be excluded in the preparatory phase of the documentation and managed as not feasible (cf. section 4.7).

Data collecting means can be used to further reduce the documentation effort required for the code rules and the variants. As an example, the code rules can be computer generated in the preparatory documentation phase based on the codes that affect a position. For the writing and assignment of the code rules to the position variants, users can be supported actively by using automatic completion functions.

The consistent use of short code rules with positive expressions allows for a complete documentation of all product variants of a product group[3]. Since the number and complexity of the code rules (bottom-up) increase with each hierarchical level, the target must be to document the code rules on the lowest level of the product structure. This fact leads to the demand to document a product web based on the smallest design elements. The following sections will address the elements of such a web as well as its structure and maintenance in detail.

4.3.4 Web of smallest design elements

The integrated product and process data model according to the interconnections documentation method manages the web of the smallest design elements (SDE) on the bottom level of the product structure. SDEs are defined as follows:

---

1 Due to reasons of technical or marketing feasibility, for example.

2 If 2 of n codes cannot be combined, 2^{n-2} of the 2^n code rules can be excluded.

3 This type of documentation documents a position variant which does not need a part at a defined position for a specific vehicle configuration as independent variant which does not reference a part to be used. The type of position variant used in this case is called "No part variant".

> A smallest design element in the sense of the interconnections documentation is an element whose design features, ports and functions are known (or specified) to the manufacturing company. The (single) parts, geometries, functions and interconnections, however, are not known.
>
> *The smallest design elements form the lowest level (= basic level) of the interconnections documentation.*
>
> *For the manufacturing process, the internal design depth is regarded as basic level. For the product engineering process, basic level means the (design) features.*
>
> *Ultimately, the smallest design elements are the individual geometry elements or (design) features.*

Taking into account the SDEs, the product web is also referred to as SDE web or basic web. The structuring elements of the SDE web are the part and interconnection positions defined above (cf. Fig. 4.10).

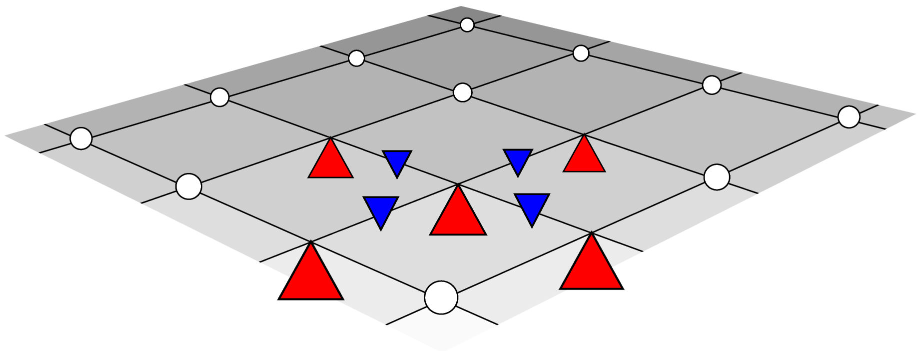


Fig. 4.10: SDE web made up of part and interconnection positions

In principle, there is a single level relation between the product and the part/interconnection positions.

4.3.5 Configuration of the basic web

In sections 4.3.1 to 4.3.4

- part positions and part position variants
- interconnection positions and interconnection position variants
- short code rules with positive expressions

have been defined as the key components and structuring elements of the basic web. This section now addresses the configuration of the SDE web in detail.

According to the interconnection represented in Fig. 4.11 as example, exactly one part or interconnection position variant (the basic version variant or "series") exists for each part or interconnection position of the basic web at the beginning of each product configuration[1].

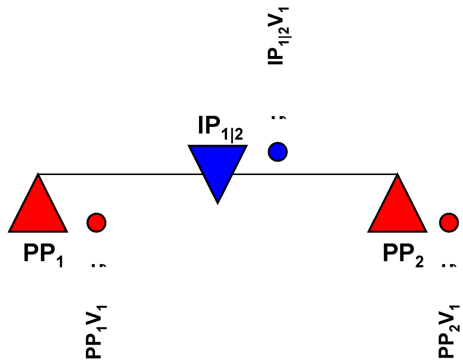


Fig. 4.11: Initial situation of SDE web

If another part position variant PP_1V_2 has to be configured for part position PP_1 in the course of product lifecycle to reflect an additional equipment variant with code C1[2], the potential effects have to be identified and evaluated. Two basic cases are distinguished.

Case 1: No additional part position variant for part position PP_2 (cf. Fig. 4.12 quadrant 1.1/1.2)

In this case, part position PP_2 remains unchanged. It is not necessary to configure a new part position variant for position PP_2. For a vehicle configuration with code C1, the basic version variant PP_2V_1 is selected for part position PP_2. Based on the interconnection $IP_{1|2}V_1$ which exists for part position variants PP_1V_2 and PP_2V_1, the effects on the processes have to be evaluated.Again, two cases have to be distinguished:

Case 1.1: No new interconnection position variant for interconnection position $IP_{1|2}$ (cf. Fig. 4.12 quadrant 1.1)

For vehicle configurations with code C1, the processes are not affected. Hence, the standard process can be applied for these vehicle configurations.

Case 1.2: New interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ (cf. Fig. 4.12 quadrant 1.2)

For vehicle configurations with code C1, the processes along the product generation process are affected. These effects are documented via an additional interconnection position variant $IP_{1|2}V_2$ of the interconnection position $IP_{1|2}$ with code rule C1. Vehicle configurations with code C1 on position PP_1 are not affected by changes or new developments on the SDE level (assumption for case 1). However, the processes might change and can be documented. The interconnections documentation can make these effects transparent and rateable. In this way,

1 In accordance with the ports & connections approach (cf. [Jüngst04]), the series variant is designed to the end.

2 Due to a new sales requirement, for example.

decision-making processes can be supported and documented in a comprehensible way (for future projects).

This completes the remarks on the assumption that the vehicle configuration does not require an additional part position variant (case 1, see above).

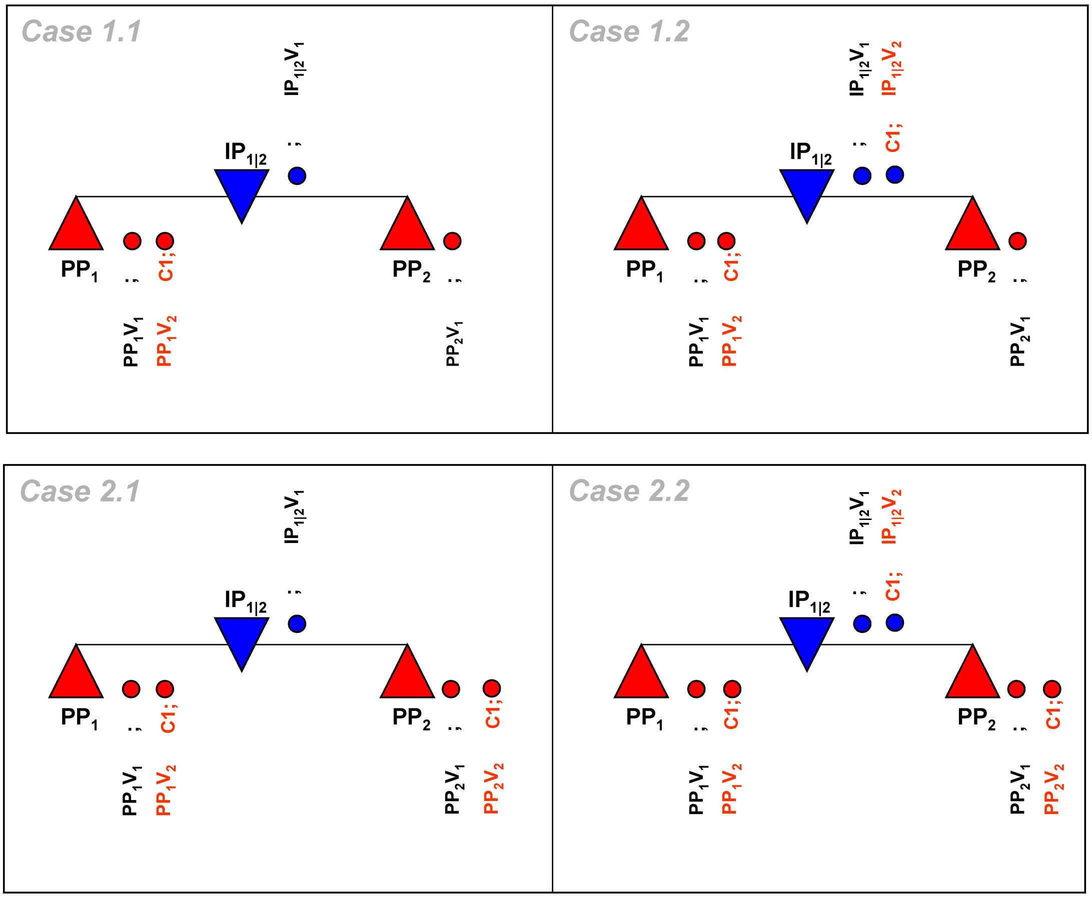


Fig. 4.12: Potential configurations to consider the effects of an additional code C1

Case 2: New part position variant PP_2V_2 for part position PP_2 (cf. Fig. 4.12 quadrant 2.1/2.2)

For PP_2, an additional variant PP_2V_2 with selection code C1 shall be configured. For the resulting combination of the part position variants PP_1V_2 and PP_2V_2, the effects on the processes must be evaluated. Again, different cases have to be distinguished.

Case 2.1: No new interconnection position variant for interconnection position $IP_{1|2}$ (cf. Fig. 4.12 quadrant 2.1)

Case 2.2: New interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ (cf. Fig. 4.12 quadrant 2.2)

For $IP_{1|2}$, an additional interconnection position variant $IP_{1|2}V_2$ with selection code C1 has to be configured.

Now, the methodological procedure for the configuration of an additional code C1 based on the initial situation of the SDE web (cf. Fig. 4.12) is described completely without loss of generality.

Due to additional requirements during the product lifecycle, further adaptations to the configuration of the SDE web may become necessary. Consequently, the effects of another equipment feature with code C2 on the SDE web have to be considered on the product configuration side without loss of generality. The part positions PP_2 and PP_1 shall be reviewed with respect to the configuration of an additional part position variant PP_2V_2 (see case 3) and PP_1V_3 (see case 4). Allowing for the statements on configuring the SDE web given so far, this expanded consideration (based on the configuration result[1] described in case 1.1) leads to the discussion of a few interesting questions. For a detailed description, the subject matter shall be differentiated further to start with.

Case 3: No new part position variant for part position PP_1 (cf. Fig. 4.13)

Part position variant PP_2V_2 with code rule C1 to be configured for part position PP_2 does not require an additional part position variant for part position PP_1. To assess the effects on the processes, however, the combinations of the part position variants already configured have to be reviewed in detail.

For the combination of the two part position variants PP_2V_2 and PP_1V_1 of the part positions PP_2 and PP_1, two other cases have to be distinguished.

Case 3.1: No new interconnection position variant for interconnection position $IP_{1|2}$ (cf. Fig. 4.13 quadrant 3.1)

For $IP_{1|2}$, no additional interconnection position variant $IP_{1|2}V_2$ has to be configured for code C2.

Case 3.2: New interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ (cf. Fig. 4.13 quadrant 3.2)

For $IP_{1|2}$, an additional interconnection position variant $IP_{1|2}V_2$ with selection code C2 has to be configured. The additional interconnection position variant makes the process variance transparent which might be generated. An evaluation of time and costs as seen from process aspects can be performed now. Based on these evaluations, the corresponding decisions can be made and documented. The interconnection objects (interconnection position and interconnection position variant) are used to document the information unambiguously and without redundancy. Experience gained can be documented as well as already available knowledge about a specific situation.

For part positions PP_2 and PP_1, the combination of the two part position variants PP_2V_2 and PP_1V_2 has to be reviewed in detail. The following cases can be distinguished:

1 The other configuration results for C1 (cf. Fig. 4.12) as initial situation for the further investigation are due to the differentiation of cases described already. Therefore, they are not a matter of discussion here.

Case 3.3: No new interconnection position variant for interconnection position $IP_{1|2}$ (cf. Fig. 4.13 quadrant 3.3)

It is not necessary to configure an additional interconnection position variant $IP_{1|2}V_2$ for interconnection position IP $_{1|2}$ for the codes C1 and C2.

Case 3.4: New interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ (cf. Fig. 4.13 quadrant 3.4)

An additional interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ has to be configured for the code rule C1+C2.

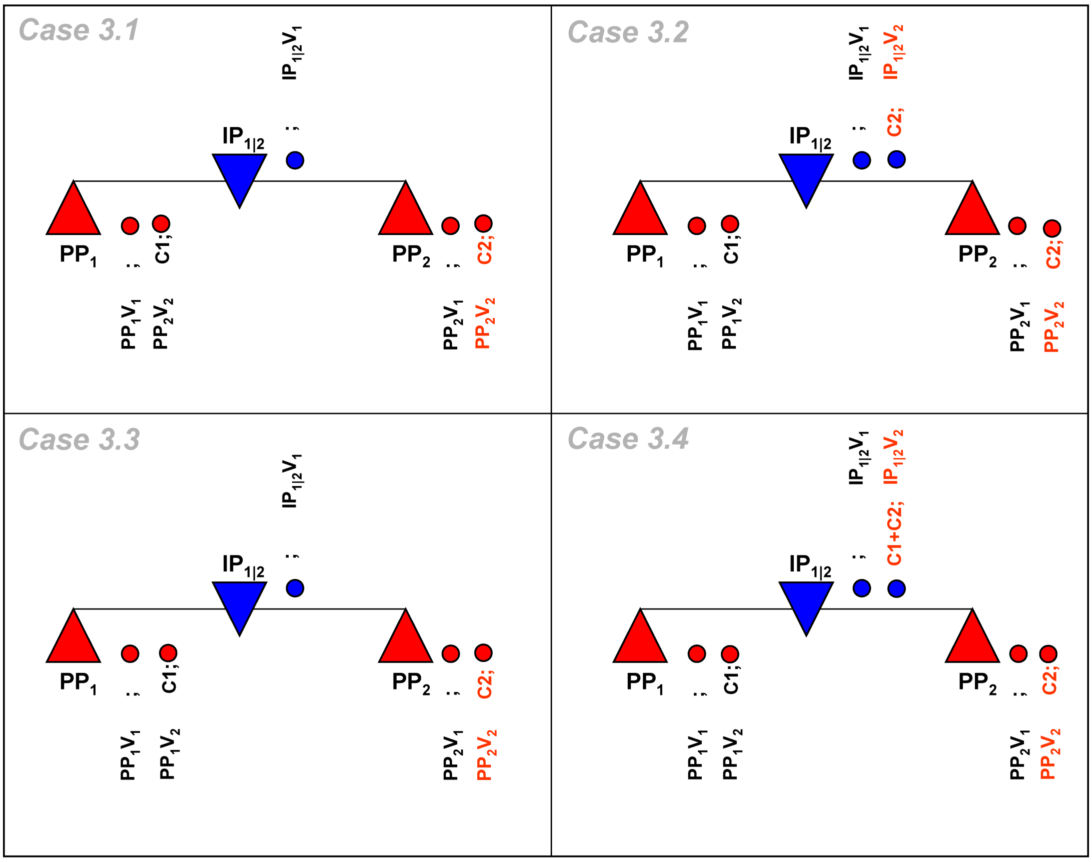


Fig. 4.13: Potential configurations to consider
the effects of an additional code C2

Finally, the case shall be reviewed where an additional part position variant PP_1V_3 shall be considered for part position PP_1.

Case 4: New part position variant PP_1V_3 for part position PP_1 (based on the configuration results of Fig. 4.12 quadrant 1.1)

For part position PP_1, a new part position variant PP_1V_3 has to be configured with code rule C2. For the combination of the two part position variants PP_2V_2 and PP_1V_3 of the part positions PP_2 and PP_1, two cases have to be distinguished.

Case 4.1: No new interconnection position variant for interconnection position $IP_{1|2}$ (cf. Fig. 4.14 quadrant 4.1)

It is not necessary to configure an additional interconnection position variant for interconnection position $IP_{1|2}$ for code C2.

Case 4.2: New interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ (cf. Fig. 4.14 quadrant 4.2)

An additional interconnection position variant $IP_{1|2}V_2$ for interconnection position $IP_{1|2}$ has to be configured for the code rule C2.

The following remarks apply to the product configuration for case 4:

- Part position variants PP_2V_2 and PP_1V_3 are always used together in those vehicle configurations which contain code C2, but not C1.
- Part position variants PP_2V_1 and PP_1V_1 or PP_1V_2 and PP_2V_1 are always used together in those vehicle configurations which do not contain code C2.
- Part position variants PP_2V_2 and PP_1V_1 or PP_2V_1 and PP_1V_3 cannot be used together, if vehicle configurations contain code C2.
- Besides, a part position variant has to be considered for part position PP_1 for code rule C1+C2.

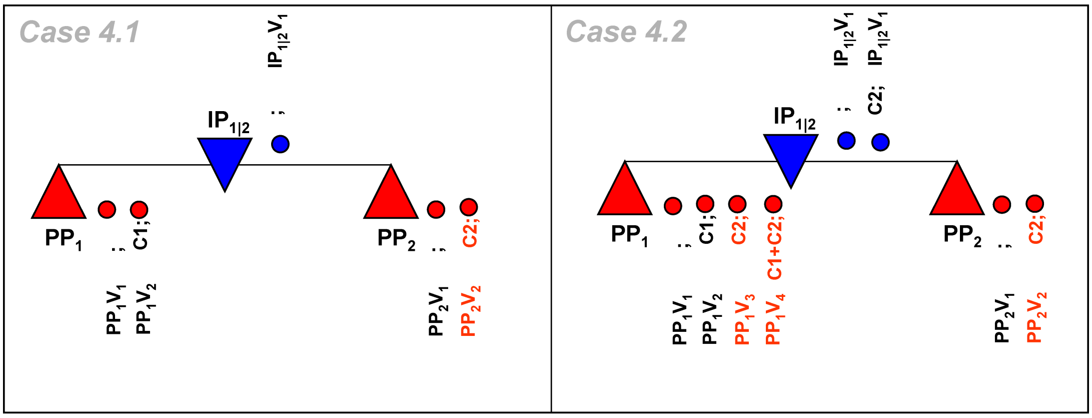


Fig. 4.14: Survey of possible alternatives
for an additional part position variant

The product configuration of the SDE web based on part positions and interconnection positions and their variants described in this section represents the core piece of the interconnection based product and process documentation.

4.4 Decomposition

As an alternative to derive the psPS from the OPS, this part of the study describes the decomposition which is a supplementing concept to the interconnection based product structuring. While section 4.4.1 addresses the necessity of this approach, section 4.4.2 describes the method of decomposition. Section 4.4.3 deals with the integration of indirect process functions.

4.4.1 Alternative methods to derive a product structure

Starting from the project description, a product is created in the product generation process in accordance with the top-down procedure (cf. section 4.1.2). Each step of splitting up creates a more detailed web of parts and interconnections

The concept of OPS and psPS described in section 4.2 represents a tool which allows for a standardized procedure to derive specific product structures and the reuse of components. Since the existence of an OPS cannot be assumed for the implementation of the interconnections documentation and the level of detailing might not be sufficient, the method of decomposition described in this section represents an alternative to standardize and facilitate the (manual) splitting up of the product web.

Thus, the decomposition can be used whenever an OPS does not exist or when a product or (existing) product structure shall be refined further The demand to detail an existing product web down to the level of the SDEs (cf. section 4.3.4) particularly results from the presently existing product complexity.

4.4.2 Methodological procedure of the decomposition

The methodological approach of decomposition presents a procedure to document a product configuration as described in section 4.3.5 based on the SDEs.

The procedure of decomposition is used when a further detailing of the product structure or refining of the existing product web is required in the product engineering process. Fig. 4.15 shows that the decomposition splits up a part position node to be detailed into a partial web. The position describing information of the product structure node to be decomposed is inherited completely to the partial web to be configured. This position describing information can include all data collected along the product lifecycle, like e.g. costs, time, resources, weight, etc.

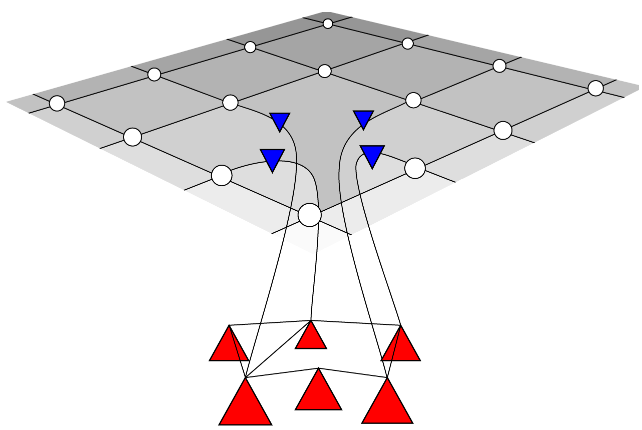


Fig. 4.15: Procedure of decomposing

To inherit the information of the initial node to be decomposed to the different part and interconnection positions of the partial web, freely definable algorithms can be used as supporting measure. This way, information can be inherited in a standardized, function specific and computer aided way.

A key characteristic of the methodological concept of decomposition is that already existing variants of the structuring node to be decomposed are entirely deleted, and their information is transferred to the newly generated partial web. This is an important requirement for the assembly or bottom-up generation described in section 4.5 and 4.8.

Another requirement placed on a successful decomposition is that all interconnection positions of the initial node are assigned completely in the new partial web (cf. Fig. 4.15).

4.4.3 Integrating indirect process functions

Frequently, processes and functions like, for example, predevelopment, technological developments or the development and manufacturing of external products occur parallelly or upstream to the product engineering process. The results and processes of such organizations, called indirect process functions, are also mapped in the interconnection based product and process model. The resulting partial product webs either can be transferred to the OPS to be available to all subsequent products and product structures, or they are integrated and configured into the "entire web" (of a specific product structure) in accordance with the decomposition method for a structuring node to be split up.

4.5 Sequencing and assembly generation

With the sequencing of interconnections, section 4.5.1 introduces another key concept of the interconnections documentation. In this context, sections 4.5.2 and 4.5.3 define the preference graph and manufacturing sequence to start with. It is followed by a description of other areas where to use sequencing given in section 4.5.4. The assembly generation introduced in section 4.5.5 represents a methodological approach which starts from the information of the preference graph or manufacturing sequence. An analysis of the effort for the number of key terms of the interconnections documentation in section 4.5.6 concludes this part of the study.

4.5.1 Sequencing the interconnections

If the definition of the interconnection position described in section 4.3.2 is expanded by an interconnection number or sequence edge, a sequence can be established between the interconnections (cf. Fig. 4.16). Different interconnections with an identical interconnection number can be performed at the same time (i.e. parallelly). Interconnections with different interconnection numbers have to be performed one after another. In principle, the interconnections with a smaller interconnection number are performed prior to those with a higher interconnection number.

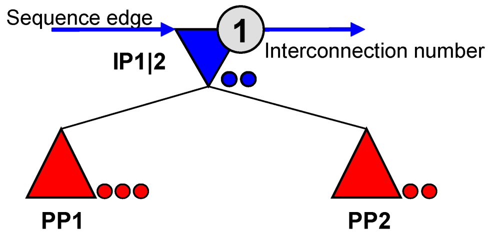


Fig. 4.16: Sequencing interconnections via interconnections number and sequence edge

The detailing of sequencing in the design that certain interconnections have to be begun, but must not have been completed before the next interconnection can be performed shall be neglected here on purpose. However, the general concept of sequencing can also be applied to these cases.

4.5.2 Preference graph

The sequencing of the interconnection positions can be used to determine the preference graph on the level of the SDEs. The preference graph is defined as follows:

> A preference graph is defined as the theoretical assembling sequence of a product from a purely design related view using predecessor/successor relations (cf. Fig. 4.17).
>
> *The designer defines the degrees of freedom of the preference graph.*

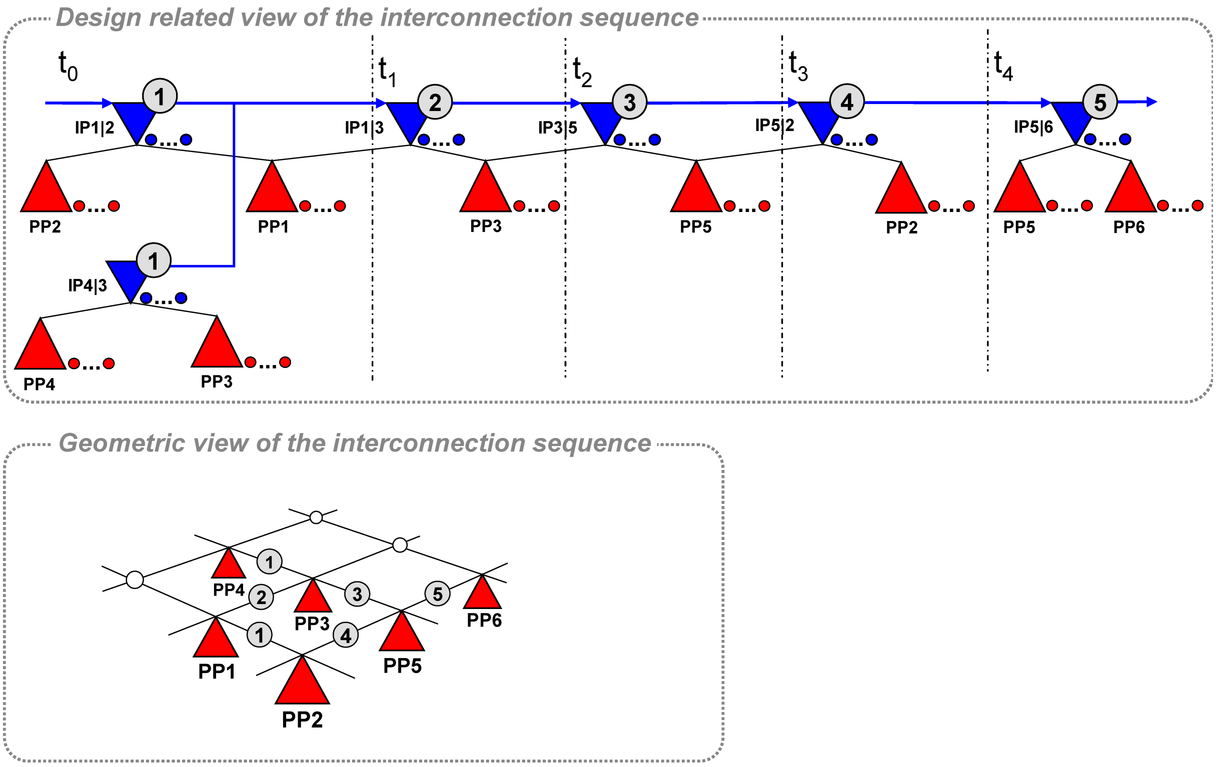


Fig. 4.17: Preference graph

4.5.3 Manufacturing sequence

In contrast to the preference graph, the sequencing of the interconnection positions plays a more important role for the manufacturing sequence. Sequencing results from the restrictions which occur from the factory or production situation. The manufacturing sequence can be defined as follows:

> The manufacturing sequence describes the sequence which results from the requirements of production regarding the manufacturing procedures (e.g. automation) and factory planning (e.g. line assignment).
>
> *The manufacturing sequence restricted by production must be within the restrictions given by design preference graph (cf. Fig. 4.18).*

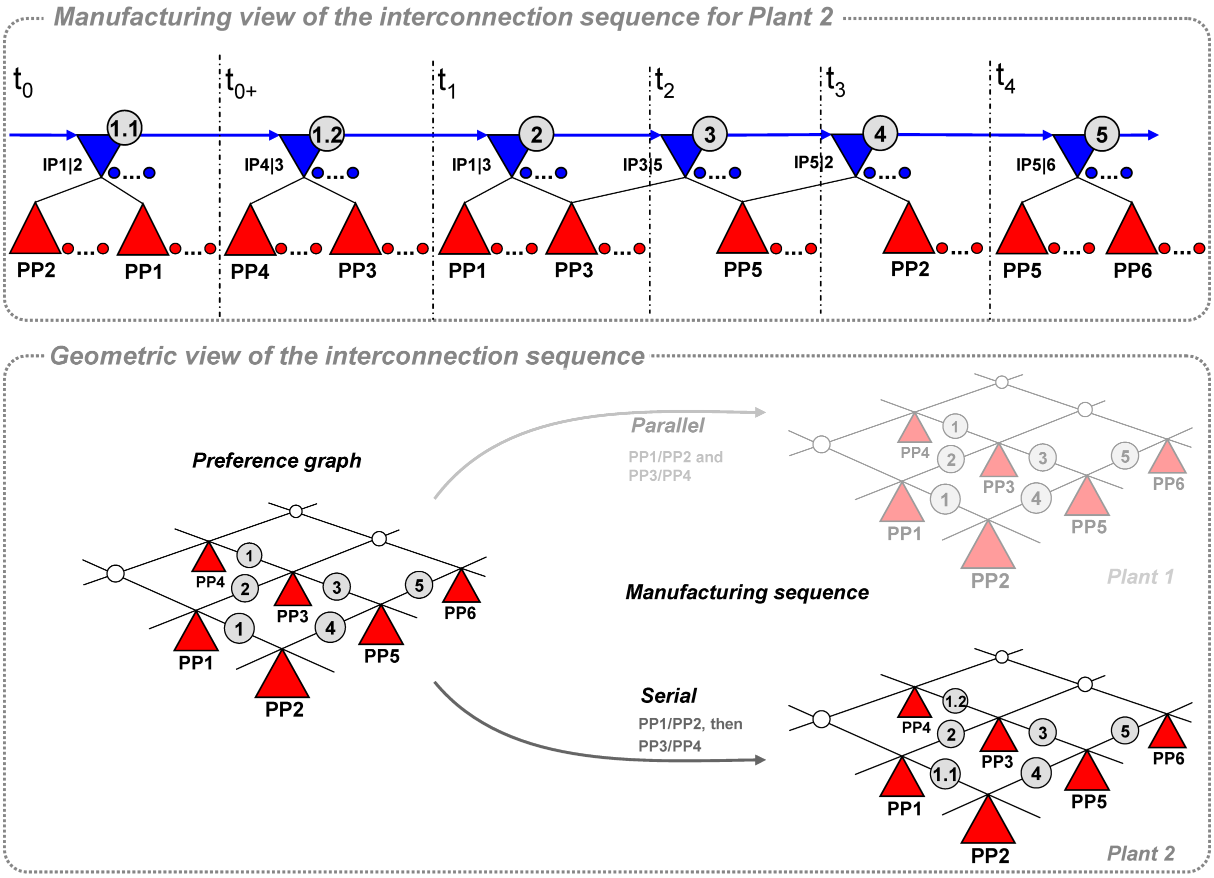


Fig. 4.18: Manufacturing sequence

4.5.4 Other applications of sequencing

Apart from the applications of the concept of sequencing interconnection positions in the form of the preference graph or manufacturing sequence presented here, the concept can also be transferred to other business cases. Two examples shall be mentioned here:

- Documenting the disassembling (e.g. in recycling, after sales, or customer service).
- Documenting the assembling instructions for optional or temporarily valid vehicle configurations which can be performed by the customers on their own. These instructions can be created, for example, for add-on parts or the replacement of specific parts (change exterior mirror to a different color e.g.) during the product usage.

4.5.5 Assembly generation

Sections 4.5.2 and 4.5.3 introduced the sequencing of interconnection positions in the context of preference graph and manufacturing sequence. This section now will explain the subsequent assembly generation.

The interconnections documentation manages all product and process specific information in the form of the SDE web (cf. section 4.3.4) on the bottom level of the product structure. Along the product generation process, nevertheless, there is the frequent demand to obtain information about the variants of an assembly at defined positions in the process.

The previously defined positions model of an interconnection can be expanded by the assembly position and its variants (cf. Fig. 4.19 a) and defined as follows:

> An assembly position represents the results of an interconnection (between two part positions) or a series of interconnections (cf. Fig. 4.19 b). Hence, there is a direct relation to the interconnection position.
>
> *(As for the part position), an assembly position is represented with the graphical element of a (red) rectangle.*

> Assembly position variants are created by generating the part positions or their variants which belong to an interconnection or series of interconnections.
>
> *Assembly variants can also be differentiated by non-product describing position variants (e.g. differentiation by different process/procedure variants). A concrete variant is selected via the code rules documented at the position variants.*
>
> *Assembly variants are represented with the graphical element of a (red) circle.*

The conventional direct structural relation between assembly and single part does not exist any more (cf. Fig. 4.19). Each assembly, including the complete vehicle, merely is the result of an interconnection or a series of interconnections. The assembly itself is not used in interconnections. The single part level of the smallest design elements, ultimately that of the geometry elements, forms the foundation.

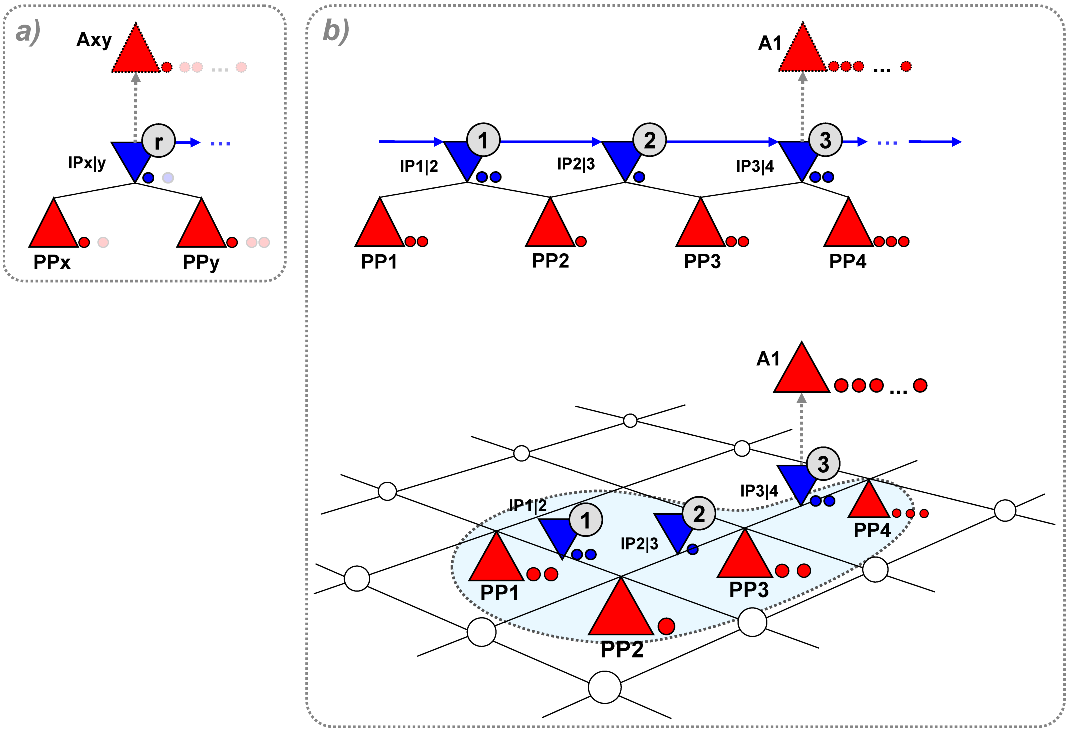


Fig. 4.19: Assembly generation

With the sequencing of the interconnection positions and the related documentation of the preference graph and the manufacturing sequence, the interconnections documentation provides an instrument which allows the (bottom-up) generation of assemblies anytime in the manufacturing process (cf. Fig. 4.4). As an example, the costs can be generated at any point in time.

To generate the assemblies, the interconnection positions which limit the assembly simply have to be identified (cf. Fig. 4.19). These interconnection positions define the content and volume of the assembly to be generated. The interconnection positions to limit the assembly are not restricted in their selection sequence. In contrast to the conventional documentation methods where the assemblies have to be defined in the run-up, the interconnections documentation allows a dynamic generation of the assembly variants anytime.

This concept of generating assemblies offered by the interconnections documentation represents a decisive advantage compared to the conventional documentation procedures. It is with this complete documentation of all product and process specific information on the level of the SDE web that the complete variance of assemblies can be provided anytime in the product generation process[1].

1 This applies both to first and second level completeness (cf. section 1.1.1).

4.5.6 Analysis of the number of key terms

At this point, it shall be stated explicitly that it is only by considering the sequencing of interconnections as intended by the preference graph and the manufacturing sequence that the number of key terms (cf. section 4.3.1 and 4.3.2) can be determined completely and compared to those of a conventional (non-interconnection based) product and process documentation.

The following is true for the number of key terms[1]:

- The interconnections documentation method requires, as maximum, as many terms as today's (conventional) documentation methods (even in the simplest case, i.e. products without variants).
- In contrast to today's (conventional) documentation methods, the number of key terms in the interconnections documentation is reduced reciprocally to product variance.

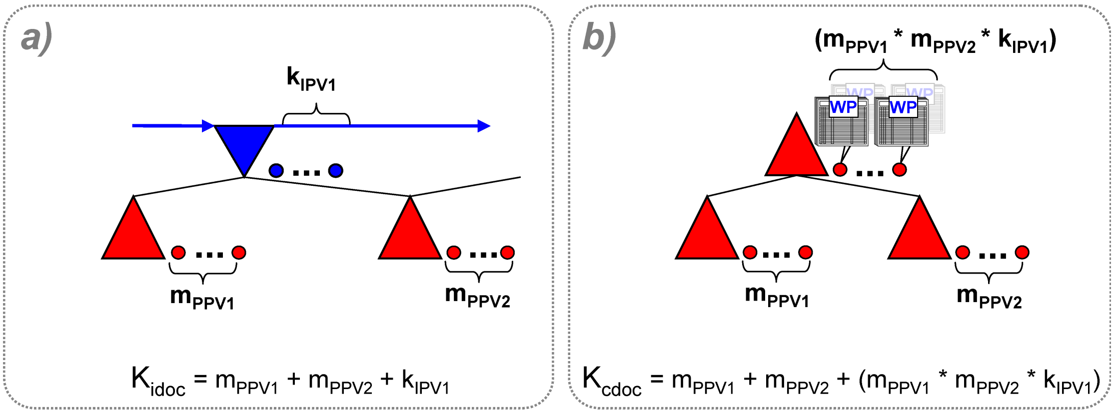


Fig. 4.20: Number of key terms in a comparison

For the analysis of the number of key terms, the following consideration can be put up (cf. [Bedi08]):

For a simple interconnection consisting of two part positions and an interconnection position, the documentation effort is as follows (cf. Fig. 4.20 a):

$$K_{idoc} = m_{PPV1} + m_{PPV2} + k_{IPV} \quad (1)$$

K_{idoc} = total number of key terms to be documented for the interconnections documentation; for m_{PPV1}, m_{PPV2}, k_{IPV1} >= 1; m_{PPV1} or m_{PPV2} represents the number of variants of part position PP_1 or PP_2. Similar, k_{IPV1} means the number of interconnection methods of interconnection position $IP_{1|2}$.

To map the same content of information via the conventional documentation method, an assembly has to be documented in accordance with Fig. 4.20 b) (cf. also section 2.5.3 or Fig. 2.15). For the number of key terms in accordance with the conventional method, the effort is as follows:

$$K_{cdoc} = m_{PPV1} + m_{PPV2} + (m_{PPV1} * m_{PPV2} * k_{IPV1}) \quad (2)$$

1 It is up to the user to prove these hypotheses (inductive proof).

K_{cdoc} = total number of key terms to be documented for conventional documentation method; for m_{PPV1}, m_{PPV2}, k_{IPV1} >= 1; m_{PPV1} or m_{PPV2} represents the number of variants of part position PP_1 or PP_2. In the same way, k_{IPV1} is the number of interconnection methods.

With an inductive proof, the hypotheses stated above shall be proven without loss of generality:

$$K^n_{idoc} = < K^n_{cdoc} \qquad (3)$$

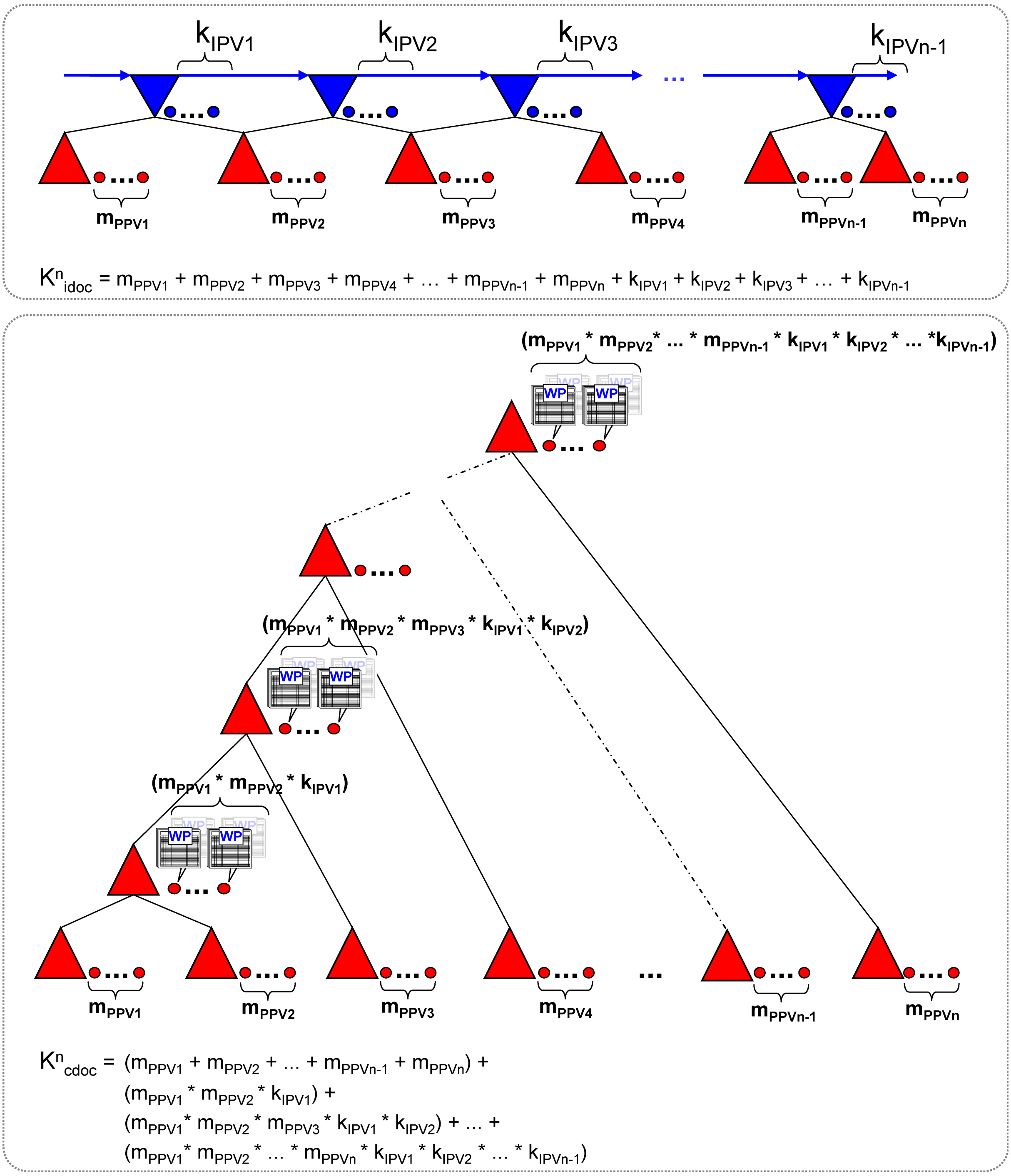


Fig. 4.21: Number of key terms in a comparison

Considering the insights for (1) and (2), the following inequality is set up for n part positions (cf. Fig. 4.21):

$$K(n): \quad (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1} + m_{PPVn}) + (k_{IPV1} + k_{IPV2} + ... + k_{IPVn-2} + k_{IPVn-1}) =< (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1} + m_{PPVn}) + (m_{PPV1} * m_{PPV2} * k_{IPV1}) + (m_{PPV1} * m_{PPV2} * m_{PPV3} * k_{IPV1} * k_{IPV2}) + ... + (m_{PPV1} * m_{PPV2} * ... * m_{PPVn} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-1}) \quad (4)$$

For two part positions and k_{IPV1} interconnection methods, the following is true:

$$K(2): \quad m_{PPV1} + m_{PPV2} + k_{IPV1} =< m_{PPV1} + m_{PPV2} + (m_{PPV1} * m_{PPV2} * k_{IPV1}) \quad (5)$$

$$k_{IPV1} =< m_{PPV1} * m_{PPV2} * k_{IPV1}$$

$$1 =< m_{PPV1} * m_{PPV2} \quad (6)$$

This is true for all m_{PPV1}, $m_{PPV2} >= 1$.

In accordance with the inductive proof, the inequality according to (4) is true for n-1 part positions.

$$K(n-1): \quad (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1}) + (k_{IPV1} + k_{IPV2} + ... + k_{IPVn-2}) =< (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1}) + (m_{PPV1} * m_{PPV2} * k_{IPV1}) + (m_{PPV1} * m_{PPV2} * m_{PPV3} * k_{IPV1} * k_{IPV2}) + ... + (m_{PPV1} * m_{PPV2} * ... * m_{PPVn-1} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-2}) \quad (7)$$

In accordance with (3), the following notation shall be used for K(n-1):

$$K^{n-1}_{idoc} =< K^{n-1}_{cdoc} \quad (8)$$

The following shall be proven for n part positions:

$$K(n): \quad (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1}) + (k_{IPV1} + k_{IPV2} + ... + k_{IPVn-2}) + m_{PPVn} + k_{IPVn-1} =< (m_{PPV1} + m_{PPV2} + ... + m_{PPVn-1}) + (m_{PPV1} * m_{PPV2} * k_{IPV1}) + (m_{PPV1} * m_{PPV2} * m_{PPV3} * k_{IPV1} * k_{IPV2}) + ... + (m_{PPV1} * m_{PPV2} * ... * m_{PPVn-1} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-2}) + m_{PPVn} + (m_{PPV1} * m_{PPV2} * ... * m_{PPVn} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-1}) \quad (9)$$

By substitution (cf. 8), the following is true for the inequality according to (9):

$$K(n): \quad K^{n-1}_{idoc} + m_{PPVn} + k_{IPVn-1} =< K^{n-1}_{cdoc} + m_{PPVn} + (m_{PPV1} * m_{PPV2} * ... * m_{PPVn} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-1}) \quad (10)$$

The following is true:

$$K(n): \quad 1 =< m_{PPV1} * m_{PPV2} * ... * m_{PPVn} * k_{IPV1} * k_{IPV2} * ... * k_{IPVn-2} \quad (11)$$

This is true for all m_{PPVk}, $k_{IPVr} >= 1$.

4.6 Design depth versus manufacturing depth

This part of the study sets off the terms design depth and manufacturing depth against each other. Potential use cases are shown (cf. section 4.6.1). Afterwards, the effects on the documentation method will be described in sections 4.6.2 to 4.6.4 when distinguishing different cases.

4.6.1 Meaning for product and process documentation

The globalization of markets has had distinct effects on the situation of automotive industry during the past years. The organization and coordination of international networks, consisting of OEMs, system and part suppliers, design and marketing partners, also affect the requirements placed on a documentation method. This section addresses the effects of the multilayered and complex customer and supplier network, on the side of vehicle engineering as well as vehicle and component manufacturing, on the documentation method in the context of design and manufacturing depth.

The term of design depth describes the detailing level of the design while the term of manufacturing depth addresses the detailing level of the manufacturing process. In this context, the following situations have to be distinguished:

- manufacturing depth equals design depth (MD = DD, see section 4.6.2)
- manufacturing depth is above design depth (MD < DD, see section 4.6.3)
- manufacturing depth is below design depth (MD > DD, see section 4.6.4)

4.6.2 Manufacturing depth = design depth

The design depth corresponds to the manufacturing depth when the OEM or system supplier is solely in charge of both design and manufacturing. In this case, the interconnection based product and process documentation considers the SDE web (cf. section 4.3.4).

4.6.3 Manufacturing depth < design depth

The manufacturing depth is above the design depth when there is a customer supplier relation where the design is performed on the customer side and the manufacturing on the supplier side.

Transferred to automotive industry, it means that the OEM carries out the design, while a system supplier carries out manufacturing. In this case, the OEM internal design depth is more detailed than the OEM internal manufacturing depth. The constellation of a lower manufacturing depth in relation to the design depth can also be transferred to the OEM internal supplier relation between a major component plant and an assembling plant. In this case, the interconnection based product and process documentation considers the SDE web (cf. section 4.3.4).

Applying the interconnections documentation in a product web shared by OEM and system supplier (or engineering and manufacturing sector) offers advantages regarding

- variance control
- transparency
- representation of manufacturing progress
- supply chain

As an alternative to the documenting in a shared product web, the assemblies manufactured by a supplier can also be documented completely based on the short code rules. This corresponds to a bundling of part positions. This is only required when there is no reference to an order or the supplier does not work in the same product web.

4.6.4 Manufacturing depth > design depth

The manufacturing depth is below the design depth when a part is designed by a supplier (external design) and manufactured in-house (customer manufacturing). This situation offers two alternatives of documenting:

Alternative 1: The external designer specifies all variants (order neutral) as modular parts list including the code rules.

Alternative 2: The assemblies provided by the external designer are split up at the customer in accordance with the procedure of decomposition and represented as part positions and part position variants.

4.7 Feasibility

This part of the study deals with the meaning of feasibility in the context of the interconnections documentation. In section 4.7.1, the different feasibility types are distinguished against each other. The concept of documenting feasibility specific information in the SDE web which is introduced in section 4.7.2 provides the basis for integrating product structure and product overview.

4.7.1 Meanings of feasibility

In the following, the different meanings of the feasibility term are defined and distinguished against each other. In principle, feasibility can be differentiated by

- maturity level
- technical or sales/marketing feasibility

> The maturity level is used to define the feasibility of a position variant. The maturity level describes when a part is released to be used in the context of the product group (e.g. model range). All attributes of the position variant shall be considered.
>
> *It must be possible to reflect, for example, the following situation via the maturity level: A supplier faces problems and cannot deliver a part in time.*

The feasibility of maturity levels shall not be addressed in depth here. The topic of maturity level management is dealt with in the context of IPDM application in section 4.9.2.

> The technical or sales/marketing feasibility describes what can or cannot be built as seen from the technical point of view or sales/marketing.

The next section will present the prerequisites for an integrated documentation of the technical and sales feasibility on the foundation of the SDE web.

4.7.2 Technical feasibility

The prerequisites for a technical/sales feasibility in the context of interconnections documentation can be defined as follows according to [Monke01]:

1. Each non-feasibility is an independent interconnection position variant in the SDE web (cf. section 4.3.4).
2. An interconnection position variant has an effect from there on the implosion of all computer generated physical and functional assembly representations (cf. section 4.5.5).
3. By identifying the interconnection position variants in the SDE web, such a "non-feasibility" receives the shortest possible code rule at the same time (cf. section 4.3.3).
4. The quantity of these interconnection position variants forms the entity of all (design, manufacturing, marketing) exclusions.
5. The feasibility check represents a possibility to standardize all conditions via the laws of Boolean algebra (automated).

Using the prerequisites for a "technical/sales feasibility" introduced in this section, it is possible to derive specifications for a sales/marketing product overview based on technical restrictions.

Regarding a (vertical) integration of *Technical* and *Sales Documentation*, there are the following alternatives considering the interconnections documentation:

- Above the *Technical Documentation*, a documentation for the market remains existent (cf. [Ohl00]). The demand for this additional documentation is, amongst other, due to the existence of country specific, deviating series and optional equipment.
- For an integration of *Technical* and *Sales Documentation*, it is necessary to control national standard equipment via additional country codes. The "technical/sales feasibility" can be stored in the SDE web and generated view specific as for the product overview (cf. section 4.8).

4.8 Process specific view generation

To supplement the statements on the generation of assemblies, this part of the study gives a generally valid view of the process specific view generation based on the SDE web. While section 4.8.1 addresses the significance of the views and their manifold uses in the product generation process, section 4.8.2 explains the bottom-up generation. The model of interconnections or interconnection positions will be expanded in section 4.8.3.

4.8.1 Interconnection based views

Along the product generation process, the involved process functions have a great demand in task and function specific information.

In this context, the interconnections documentation allows to provide the task and function specific information which is required for the respective process functions based on the SDE web in the form of views. In the sense of the interconnections documentation, views are defined by bundling part and interconnection positions including their sequencing (cf. Fig. 4.22). Using bottom-up algorithms (cf. section 4.8.2), the views defined beforehand can be generated for the specific task or function for the respective process function and made available anytime.

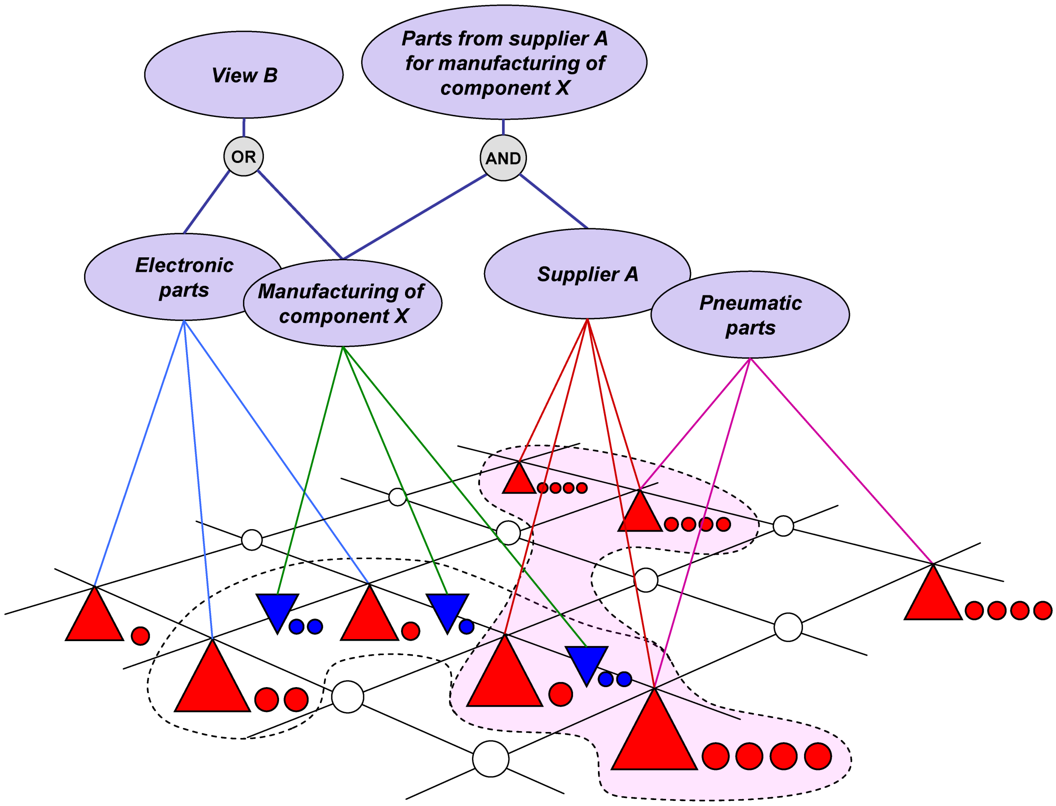


Fig. 4.22: Interconnection based views

The area of application for views is manifold. As an example, views can be used for a mere visualization of information. This type of application can be used as instrument in the decision making process. Besides, views can be used in the context of role specific workplaces. In this case, the data are not only displayed by their specific tasks, but also edited to be available to downstream process functions via the interconnection based SDE web.

4.8.2 Bottom-up generation

Using bottom-up algorithms, freely definable views can be generated for specific tasks/functions anytime along the product generation process. The following procedure steps shall be considered:

1. Identify and name the view(s).
2. Bundle part and interconnection positions (incl. sequencing) to define the view. If need be, the respective position variants can be used for restriction or differentiation.
3. Based on the code rules of the part and interconnection position variants identified in step 2, the view defined in step 1 is generated bottom-up.

The results of the bottom-up computation are provided via the appropriately defined view node. If the data provided via the view node shall be edited or modified, the modifications have to be transferred back to the SDE web to ensure that data are always up to date along the product generation process.

The method of a bottom-up generation can be performed across several hierarchical or structural levels. For the respective structural level, aggregated part and interconnection positions are provided. Using the aggregated positions, the information of the lower structural level is managed temporarily.

The bottom-up method is also used to generate the complete representation of all product variants (cf. section 4.5.5).

4.8.3 Expanded position model of an interconnection

With allowance for the information given so far on positions and their variants, and their consistent application to other business objects, the positions model of an interconnection introduced in section 4.3.2 can be expanded as shown in Fig. 4.23.

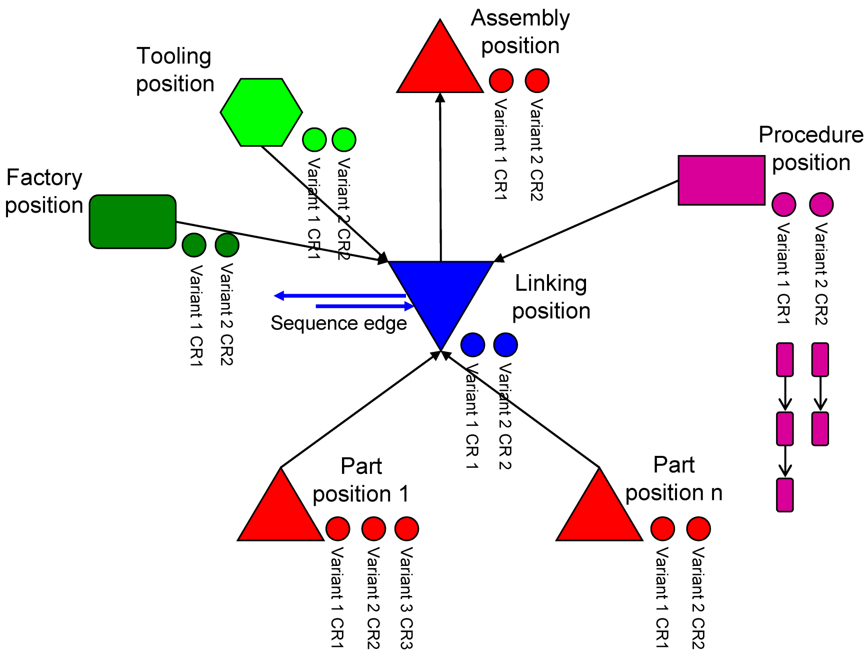


Fig. 4.23: Expanded object model of an interconnection

Apart from the part and interconnection positions, the following additional position objects with their variants can be defined:

- Procedure position and procedure variant

> A procedure position is the position related documentation of alternative procedure methods to realize an interconnection. Procedure positions are a constituent of an interconnection thus having a direct relation to the interconnection position.
>
> *Procedure positions are represented with the graphical element of a (purple) rectangle.*

A procedure variant is realized by a sequence of activities.

There is the possibility of a direct relation between activity and part position. A concrete variant is selected via the code rules documented at the procedure variant. Modifications to the procedure variants can be controlled via the interconnection position.

Procedure variants are represented with the graphical element of a (purple) circle.

- Tooling position and tooling position variant

A tooling position is the position related documentation of alternative tooling/manufacturing tools to realize an interconnection. Tooling positions are a constituent of an interconnection thus having a direct relation to the interconnection position.

Tooling positions are represented with the graphical element of a (light green) hexagon.

Tooling position variants represent concrete values of a use specific tooling position. A concrete variant is selected via the code rules documented at the position variants. Modifications to the tooling variants can be controlled via the interconnection position.

Tooling position variants are represented with the graphical element of a (light green) circle.

- Factory position and factory position variants

Factory positions (e.g. assembly line or station, line section) represent the position related documentation of alternative factory positions to realize an interconnection. Factory positions are a constituent of an interconnection thus having a direct relation to the interconnection position.

Factory positions are represented with the graphical element of a (green) rectangle with rounded corners.

Factory position variants represent concrete values of a use specific factory position. A concrete variant is selected via the code rules documented at the position variants. Modifications to the factory position variants can be controlled via the interconnection position.

Factory position variants are represented with the graphical element of a (green) circle.

The entity of all interconnection position variants is ensured by generating the maximum number of theoretical variants of an interconnection. In the sense of the interconnections documentation, the maximum number of all theoretically possible variants of an interconnection is generated by combining the position variants on the basic level of the different positions of an interconnection to be provided as assembly position *variants*.

The documentation of colors for parts can be considered a special case of an interconnection. A color position contains the color variants neutral for a part. To create the relation between a color position and a part position, the color position bundles the affected part positions similar to an interconnection position. If the color variant includes plant activities (e.g. painting), a relation via an interconnection position is also possible. The position variants of the color position are used to generate the colored variants of the single part or assembly position.

4.9 Integrated use of part and interconnection positions

The previous sections described the methodological foundation and concepts of the interconnections documentation. In the following, a few applications of the interconnections documentation shall be expanded. In this context, section 4.9.1 addresses the interconnection based change management, which is followed by the definition and structure of the integrated product and process data model (IPDM) in section 4.9.2. Based on the IPDM, section 4.9.3 presents a few applications.

4.9.1 Interconnection based change management

The change management at issue here means the management of technical change requests[1]. Essentially, the technical change management is characterized by the following process sections:

- Initializing the change request
- Evaluating the volume of changes
- Deciding on the scope of change in relation to its realization
- Realizing/implementing the scope of change

1 In contrast to this, section 4.9.3 addresses customer configurations to be changed while the vehicle has already entered manufacturing.

After initializing the change request and rating its technical feasibility, a date is determined by which the entire scope of the change shall be implemented. In the realization phase, however, it often occurs that the entire scope cannot become effective for all affected (changed and new) parts list positions at the same time. As a consequence, the original effective date for the entire undertaking (according to the weakest partial volume in relation to scheduling) has to be postponed, or the entire volume has to be split into "smallest parts list volumes which absolutely require the same effectivity date under design aspects" (= PTA[1] approach).

Although the interconnections documentation is not directly linked to the PTA approach mentioned here, it can nevertheless support or facilitate this approach. Starting from a part position or part position variant that has to be modified, the interconnections are used to identify the positions in immediate neighborhood. "Questions" may be asked where the answers will result in the forming of PTAs (cf. Fig. 4.24 a).

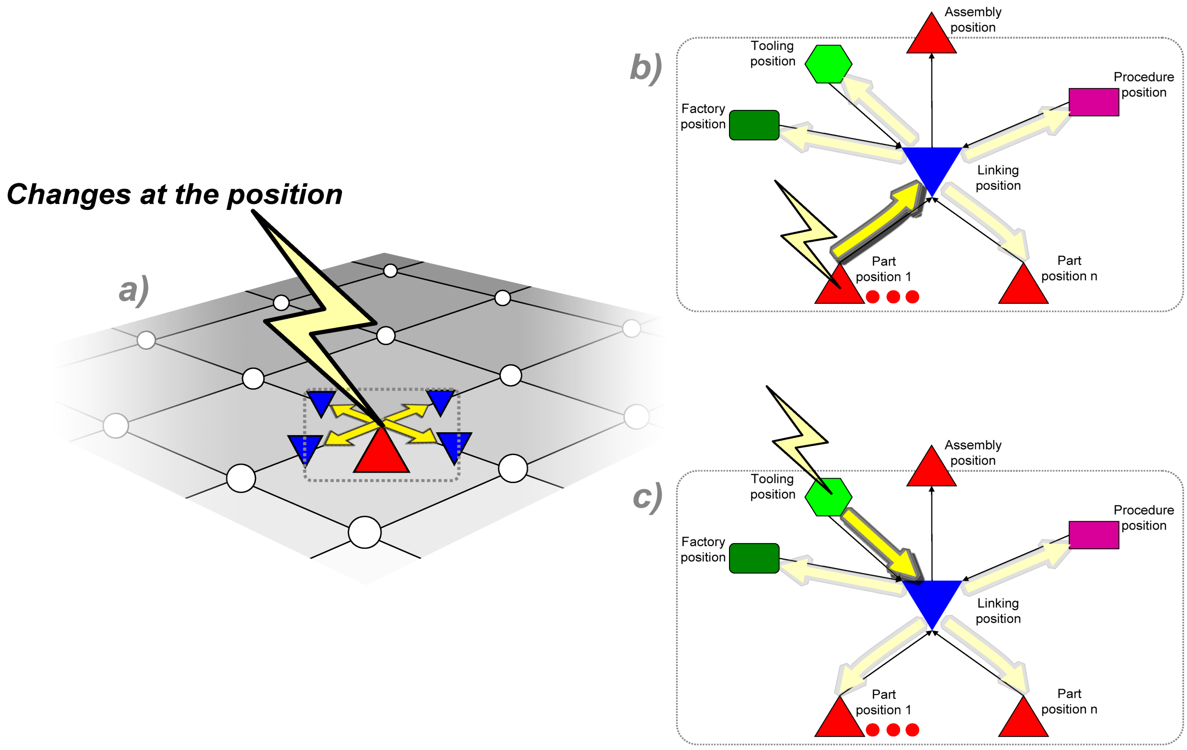


Fig. 4.24: Change management based on interconnections

Apart from helping to determine the PTA content, the method of interconnections documentation also provides the basis for an integrated change management. The interconnection positions serving as integration objects allow to identify the effects of a modification on non-part positions (cf. Fig. 4.24 b). Vice-versa, the interconnection positions also support changes of non-part positions (cf. Fig. 4.24 c). This might become necessary when, for example, a tool supplier must change a tool.

1 PTA = package of technical associativity

The essential advantages of an interconnection based change management are as follows:

- easier identification of scope of change at an improved quality
- qualitatively better rating of modifications
- improved support in decision making

The improvements in change management also have positive effects on the start-up management.

4.9.2 Definition and structure of the IPDM

For the integrated use of the part and interconnection positions, this section defines the integrated product and process data model (IPDM) and describes its structure.

The matrix of the IPDM[1] shown in Fig. 4.25 represents a logical representation of how to integrate the product generation and customer order process. The system technical realization of the logical data model can be distributed to several systems. The core piece of the IPDM and the foundation for integrating the two core processes is the SDE web defined in section 4.3.4.

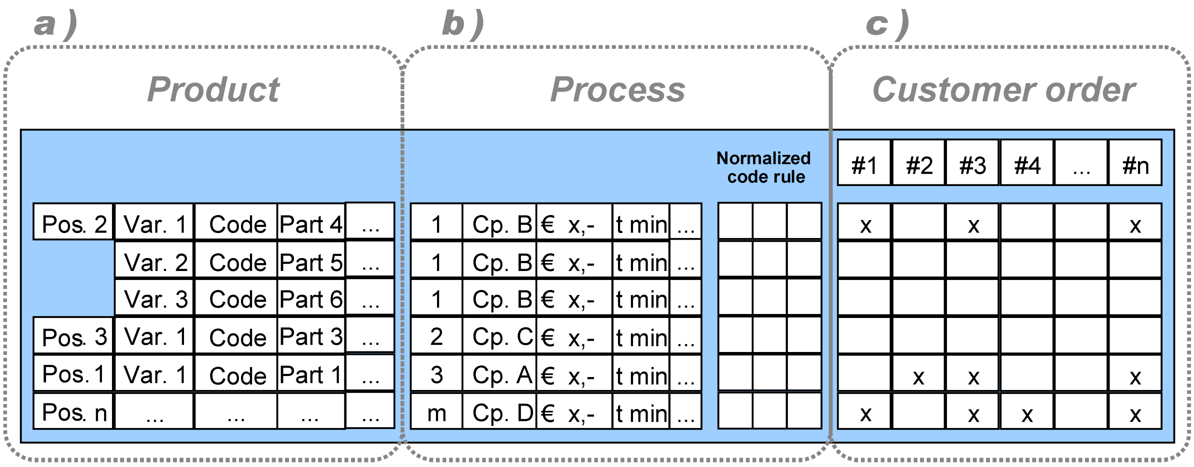


Fig. 4.25: IPDM matrix

In accordance with Fig. 4.25, the matrix of the IPDM can be divided into three areas:

- The left portion of the matrix (cf. Fig. 4.25 a) describes the complete product structure including the interconnections. Based on the key terms defined in sections 4.3.1 and 4.3.2, the rows are identified. The maturity level (cf. section 4.7.1) as constituent of the key term describes the use related status (engineering progress) of a part or interconnection in the form of a position variant. Each row thus represents a part or interconnection position variant.

1 Also referred to as IPDM matrix.

- The center portion of the matrix (cf. Fig. 4.25 b) manages the activity data (which can be processed through algorithms). All information on the process functions of all business units lead to the identification of activity data at the position variants. Activity data at the part position variants are, e.g., procurement times, minimum stock, costs, weight. Likewise, activity data for interconnection position variants are worktime, cost, weight.
- The right portion of the matrix (cf. Fig. 4.25 c) manages the customer vehicles to be manufactured in accordance with the envisaged production sequence. The vehicle orders are mirrored against the "product structure and the factory". Each column represents a vehicle order.

After explaining the structure of the IPDM matrix, the next section shall give a few examples of how to use the matrix.

4.9.3 Potential uses of the IPDM

The IPDM matrix defined above can be used to identify a few immediate use cases which will be detailed in the following:

- The demand for a customer vehicle is calculated by evaluating all part/interconnection positions and their variants considering the code rules (including effectivity dates) and vehicle configurations for the configured vehicle (cf. Fig. 4.26 a).
- The demand for a defined position variant is calculated by adding the number of individual demands of this position variant for a defined manufacturing period or number of customer orders (cf. Fig. 4.26 b).
- Based on the IPDM matrix, the order sequence can be optimized to maximize capacity exploitation or avoid existing capacity shortfalls. In the example shown in Fig. 4.27, orders #4 and #5 cannot be realized due to limited capacity on interconnection I203 for the operation Op203. To resolve the capacity restriction, order #4 is rearranged (between order #1 and #2), and operation Op203 does no longer represent a restriction infringement.

Object	Code rule	Attributes: Qty	Attributes: min	Attributes	Order with code bar / Hits	#4 C2	#1 C4	#2 C1	#3 C5	#5 C3
▲ MC203...	C2/C3;	2		3		X				X
▲ MC203...	C1;	2		5				X		
▲ MC203...	;	2		2			X		X	
▼ I203...	C2/C3;					X				X
P203...	C2/C3;		0.28	0.5		X				X
Op203...	C2/C3;		7.12	9.2		X				X
▼ I203...	C1;							X		
P203...	C1;		0.22	0.4				X		
Op203...	C1;		0.15	1.3				X		
▼ I203...	;						X		X	
Op203...	;		0.11	1.0			X		X	

b) (row Op203... C2/C3;) a) (column #2 C1)

Fig. 4.26: Demand calculation

Object	Code rule	Attributes: Qty	Attributes: min	Attributes	Order with code bar / Hits	#1 C4	#2 C1	#3 C5	#4 C2	#5 C3
▲ MC203...	C2/C3;	2		3					X	X
▲ MC203...	C1;	2		5			X			
▲ MC203...	;	2		2		X		X		
▼ I203...	C2/C3;								X	X
P203...	C2/C3;		0.28	0.5					X	X
Op203...	C2/C3;		7.12	9.2					X	X
▼ I203...	C1;						X			
P203...	C1;		0.22	0.4			X			
Op203...	C1;		0.15	1.3			X			
▼ I203...	;					X		X		
Op203...	;		0.11	1.0		X		X		

→

Order with code bar / Hits	#4 C2	#1 C4	#2 C1	#3 C5	#5 C3
	X				X
			X		
		X		X	
	X				X
	X				X
	X				X
			X		
			X		
			X		
		X		X	
		X		X	

Fig. 4.27: Optimizing order sequence

Apart from the possible applications of the IPDM matrix already described, the matrix can also be used with regard to a transparent factory in realtime:

- Production progress: Production progress can be analyzed and evaluated with date and time at any point in time.
- Target level and performance: The IPDM can be used for a transparent representation of the target level (what has been achieved) and the performance potential (what can still be achieved).

- Supplier integration: The online integration of the suppliers promotes collaboration on one's own responsibility (suppliers can react themselves towards quality problems). It offers the possibility of supplier competition for arising manufacturing/supply demands.
- Flexibility limits: The flexibility of the manufacturing plants regarding manufacturing capacities can be stored at the attributes of the (part and interconnection) position variants.
- Customized configuration changes: Customer changes regarding vehicles already configured and planned can be evaluated anytime and even considered for vehicles which have already entered manufacturing.
- Customer prioritization: Using the IPDM matrix, customer prioritization can be performed for marketing.

4.10 Migration

To implement the interconnections documentation step and step with an evolutionary realization, this section introduces a step-by-step migration concept (cf. section 4.10.1). The initial, non-reducible migration step for the implementation of the interconnections documentation is described in section 4.10.2. The further procedure of the migration process is explained in section 4.10.3.

4.10.1 Iterative migration

A successful deployment of the interconnections documentation requires a step-by-step migration. The necessity of such an iterative procedure is due to the fact that the deployment of the interconnection based product and process documentation requires both a new IT system and a new documentation method. The effects on the product structure have consequences for the entire product generation process. With the step-by-step migration of implementing the interconnections documentation in corporate practice, the risk of a failing deployment project can be reduced distinctly. The subsequent sections 4.10.2 and 4.10.3 introduce a concept for such a step-by-step migration.

4.10.2 Initial migration step

The smallest, non-divisible migration step for implementing the interconnections documentation contains the building and maintaining of the SDE web described in section 4.3.5 including the feature of determining effectivity dates (i.e. effectivity of the position variants). The condition (= code rule) and its effectivity are the prerequisites for a correct demand calculation.

Based on the SDE web and using the view concept described in section 4.8, the conventional parts list types can be derived with computer programs (cf. Fig. 4.28 a).

Deriving the conventional parts lists forms the data foundation to keep all subsequent process functions in accordance with the previous (conventional) procedures.

The realization of this first migration step produces the advantage of a complete documentation in all (part and interconnection) positions and the short code rule (cf. section 4.3).

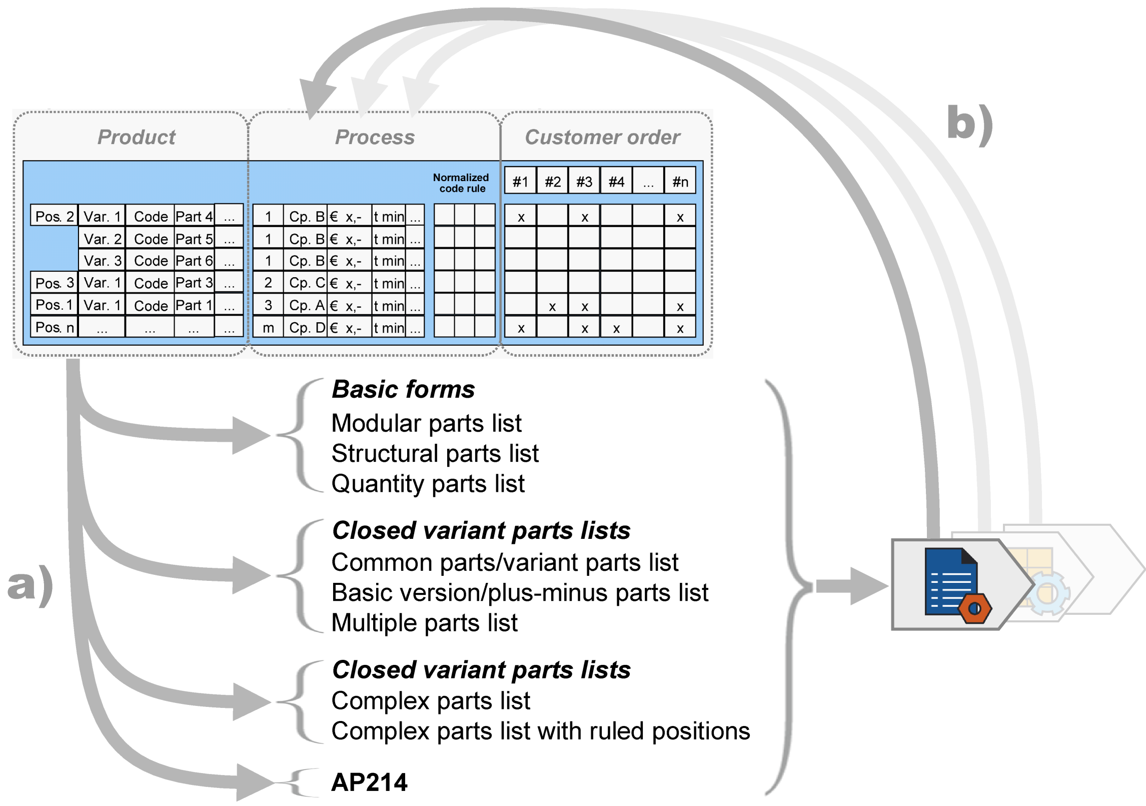


Fig. 4.28: SDE web and view generation as initial step of migration

4.10.3 Further migration procedure

When the initial migration step (see above) has been implemented successfully, the process based on interconnections can now be redesigned step by step. All process owners have to redesign the processes for the process functions they are responsible for based on interconnections. The resulting process changes affect the activity data of the center portion of the IPDM matrix (cf. Fig. 4.28 b).

For the respective process functions, it is recommended to complete the migration of the different product groups (e.g. model ranges) in a single step. This avoids the problem that different documentation methods are used simultaneously for the different product groups.

4.11 Summary

By defining and deriving the interconnections documentation, section 4.1 presented the following concepts of the interconnections documentation in an overview:

- definition of the overall product schema (OPS) and the method to derive the project specific product structure (psPS)
- building and maintaining the product web (= basic web) based on parts and interconnections
- process specific view generation via bottom-up algorithms

A detailed explanation of the substantial characteristics of these concepts was given in sections 4.2 to 4.8. In this context, section 4.2 described the definition and structure of the OPS and a methodological procedure to derive the psPS. The management of process specific information in the OPS and the organizational embedding of the OPS were explained as well.

The definitions of the part and interconnection positions, the part and interconnection position variants and the introduction of the short code rule with positive expressions in section 4.3 offered the key components of the basic web (= SDE web). With the interconnections, a new key class was introduced providing the basis for an integrated product and process data model (IPDM). In this context, a detailed methodological procedure to configure this product and process web was described.

The concept of decomposition introduced in section 4.4 offered a methodological procedure which allows for a further structuring of a product and process structure derived from the OPS or define such a structure anew.

The results of sections 4.2 to 4.4 offered a standardized procedure to develop a project specific product structure and a tool to actively manage variants and complexity based on parts and interconnections.

The introduction of sequencing interconnections in section 4.5 expanded the interconnection based product and process data model. Mapping the preference graph and the derived manufacturing sequence provided the basis for controlling the engineering process and the manufacturing sequence of a vehicle alike. Based on the complete documentation of all product and process specific information in the SDE web and the sequencing of the interconnections, this section also described the generation of assemblies. Compared to conventional documentation methods, this procedure offers the possibility of displaying the complete variance of the assemblies or the product manufactured up to that point in time anytime in the product generation process.

The mutual effects of design and manufacturing depth and their consequences on the documentation were explained in section 4.6. The process of deriving specifications for a sales/marketing product overview based on technical restrictions was described in section 4.7.

To live up to the demands of the process functions involved in the product generation process to have access to information in relation to tasks or functions, section 4.8 presented the concept of bottom-up algorithms to generate process specific views. Afterwards, the interconnection based product and process data model was expanded by defining additional positions.

While sections 4.2 to 4.8 focused on the basics of the interconnection based product and process data model and the description of the documentation method which is at the bottom of this model, section 4.9 introduced the integrated use of part and interconnection positions. At first, the conceptional approach of the change management based on interconnections was described, followed by the description of the structure and definition of the IPDM matrix giving a logical representation of the integrated product and process data model. For the integration of the customer order and product generation process realized in this way, a few use cases were described as examples.

With the concept of an iterative migration strategy and the description of the initial migration step, the structure and maintenance of the SDE web including the effectivity function (i.e. effectivity dates of the position variants), this chapter was completed.

As a consistent advancement of the BCT model (cf. section 2.4.7), the method of the interconnections documentation belongs to the class of the "open" parts list methods in the widest sense (see section 2.4.1). The expanded concept of documenting ruled positions and the related introduction of interconnections provide the prerequisite for an integrated product and process data model.

Part IV

IMPLEMENTATION

5 Design and basics of IT implementation

This chapter of the study describes the design and the basics for an IT implementation of the interconnections documentation laid out in chapter 4. At first, section 5.1 introduces the essential elements of an object oriented design considering the Unified Modeling Language. This is followed by an introduction of the architectural concept and the technological basics for an IT implementation of the interconnections documentation in section 5.2. Based on these results, section 5.3 addresses the IT technical realization of the prototype.

5.1 Unified Modeling Language

For the object oriented design of the interconnections documentation, this section introduces the basics and key diagram types (cf. section 5.1.1) of the Unified Modeling Language (UML). As a follow-up, sections 5.1.2 to 5.1.5 describe the diagrams used for the design of the interconnections documentation (class, use case, activity, sequence diagram) in detail.

5.1.1 UML basics

UML is a widely spread notation for the object oriented modeling. In principle, it is founded on three columns which are represented by the works of the Three Amigos Grady Booch, Ivar Jacobson, and James Rumbaugh. Since version 1.1, the OMG (Object Management Group) has standardized and enhanced UML (up to the current version 2.1) [Burkha99].

UML is mainly used for the modeling of object oriented software systems[1]. A model is an image of an information system with the purpose to understand the system and document the existing correlations. The system modeling takes place with different degrees of abstraction. UML is used for an abstract modeling of systems. As a universal modeling language, UML offers various diagram types which are used to model different views of a system. Depending on the used diagram type, it is appropriate to visualize workflows and structures, and present

1 Meanwhile, UML has also entered other areas of science outside informatics providing an adequate means of representing knowledge structures.

them in a clear-cut way. According to [Jeckle03], the UML diagrams can be divided into two types:

- Structure diagrams[1] are suited to model the relevant data types and structural correlations of a system.
- Behavior diagrams[2] are used when (branched) workflows, features, processes, etc. have to be represented.

Fig. 5.1 shows the 13 diagram types version 2 of the UML consists of.

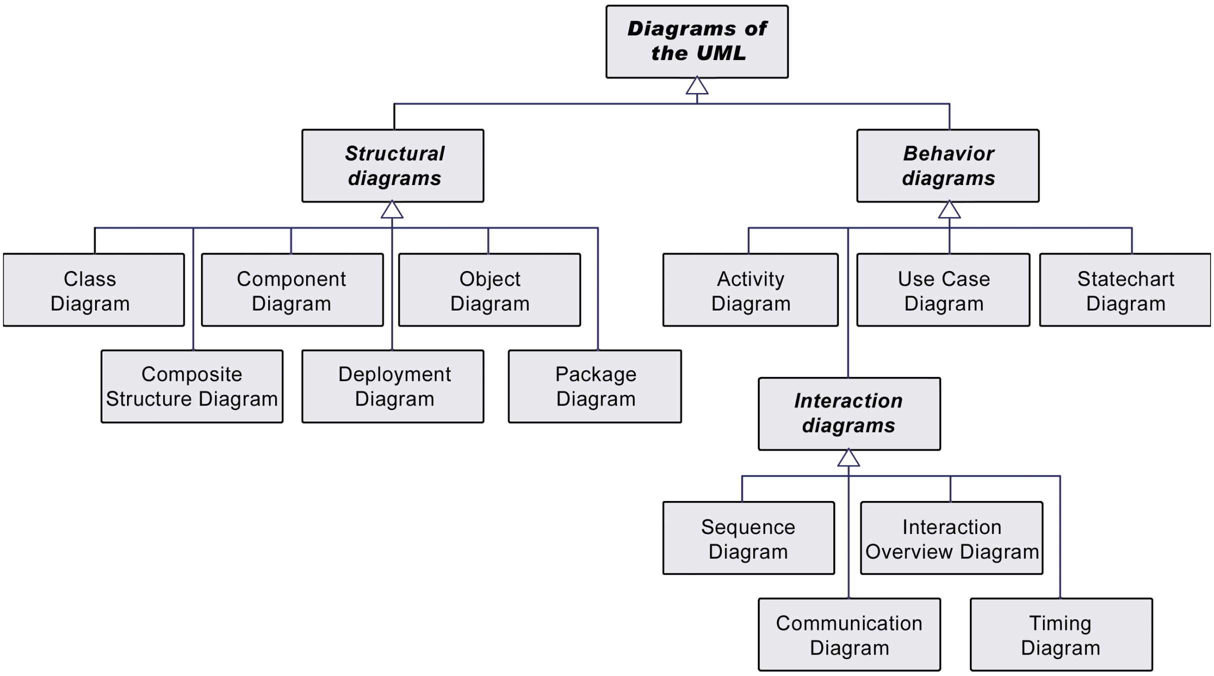


Fig. 5.1: UML diagram types (from [Jeckle03])

The syntax and semantics of UML diagrams are predefined by the OMG. In the following, those UML diagram types only will be introduced which are used in this study[3].

5.1.2 Class diagram

The class diagram belongs to the structure diagrams and describes the static structure of a system. It defines the relevant "types"/"elements"[4] a system consists of and their links to each other. Links are divided into the three categories "ordinary association", "aggregation", and "composition":

1 Class diagram, object diagram, package diagram, component diagram, composite structure diagram (new in UML 2), deployment diagram.

2 Use state diagram, state machine diagram, activity diagram, sequence diagram, communication diagram (new in UML 2), timing diagram (new in UML 2), interaction overview diagram (new in UML 2).

3 For a brief overview of all notations of the UML version 2.0 cf. [WWWu1].

4 Typically, these "elements" are classes of an object oriented programming language.

- An association is a relatively loose link between two objects. It can be linked or split as desired. Each object can be created or deleted independently. If an object is deleted, all associations where it is used are also deleted. Otherwise, the respective association partners are not changed. Hence, objects which are linked via an association are completely independent of each other.
- An aggregation describes an assignment link, to be more precise the objects which are assigned to a superior object. Objects linked via an aggregation have a stronger relation than they would have when linked via an association. Both relation partners can be created independently of each other. Aggregations can also be linked and split anytime. If a higher object (client) is deleted, all lower objects (supplier) linked via an aggregation are also deleted unless they are contained in another aggregation.
- The composition is the strongest link between two objects. It describes a "consists of" link. It specifies the parts (supplier) an entity (client) consists of. The client only can be created independently. The suppliers can only be created in the context of a client, since they are not conceivable or cannot exist without it. Besides, the suppliers belong to exactly one client. If the client is deleted, all suppliers are also deleted automatically.

Apart from these link categories, the generalization is a link type to structure the software system. The generalization describes a link between a general class (basic class) and a specialized class. The specialized class is completely consistent with the basic class, but contains additional attributes and associations [Balzer05].

The key components[1] of the class model for the interconnections documentation are shown in the class diagrams of Fig. 5.2[2] and 5.3.

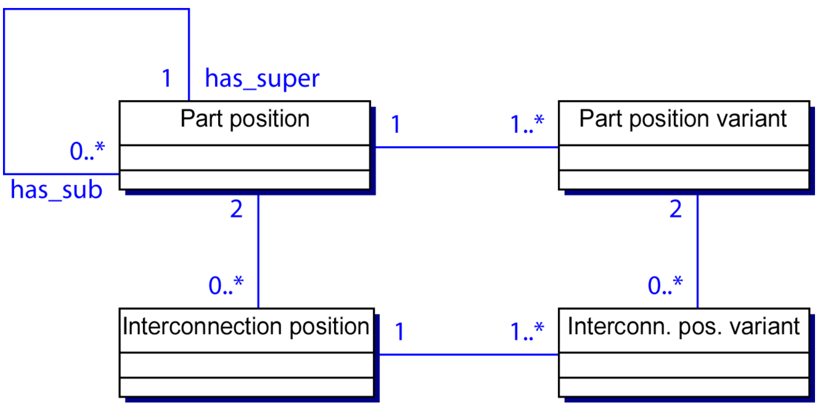


Fig. 5.2: Design of the positions of the interconnections documentation

While Fig. 5.2 focuses on the structuring of the positions and variants, Fig. 5.3 shows the substantial relations between the part and interconnection positions and their variants. The

1 The classes and their links among each other.

2 Due to missing data, a few diagrams are available in German only.

class diagrams represent the foundation of the interconnections documentation. All following diagrams and descriptions start from the design of this static model.

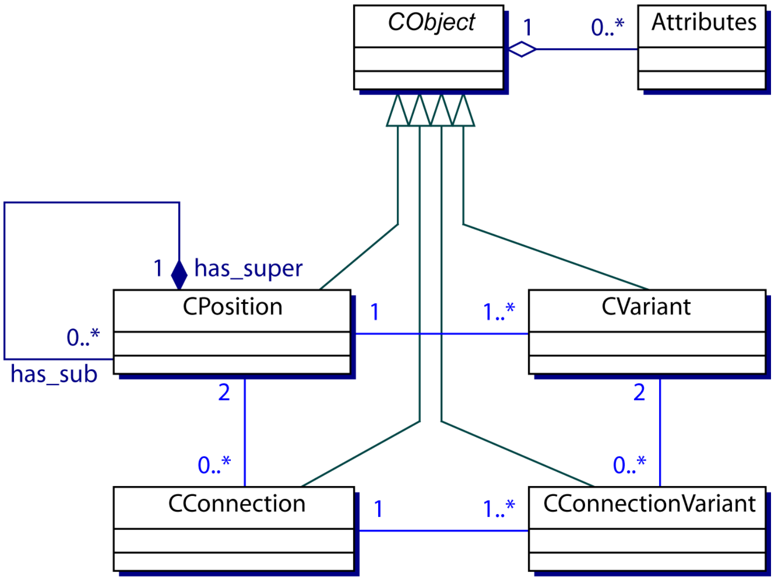


Fig. 5.3: Design of the part and interconnection positions including their variants

5.1.3 Use case diagram

The use case diagram belongs to the behavior diagrams. It is used to describe the behavior of a system or specify requirements placed on a system[1]. Simple and easily understandable means of notation are available for use case diagrams which allow for a clear delimitation of context. On a high level of abstraction, the use case diagram thus gives a good overview on the system and its interfaces to the environment [Balzer05]. The feature a user performs to obtain a desired result or achieve a target is referred to as *use case*. The quantity of all *use cases* forms the *use case* model.

An actor represents a certain role outside the system, but interacts with the system. Actors are associated with use cases via associations. The use cases triggered by an actor must produce results which are visible to the actor.

The behavior of a *use case* can be expanded by another use case via an *extend* relation. This allows for the modeling of hierarchical behavior units. An *include* relation is used to indicate that the behavior of a use case is contained entirely in another one.

Fig. 5.4 shows the most important use cases of the interconnections documentation. A structuring of the use cases starting from the use cases for creating product and process structure and performing product and process configuration can be recognized.

1 With the use case diagram, it is not possible to describe flows as is the case, for example, with a sequence or activity diagram.

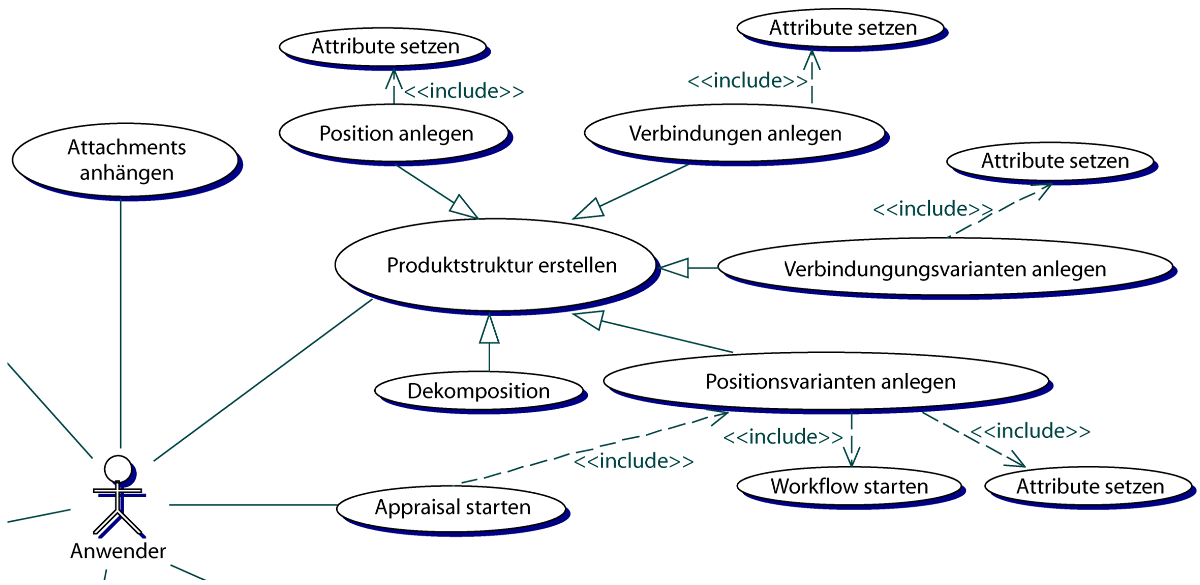


Fig. 5.4: Use case diagram of the interconnections documentation

Via the *extend* relation, the two basic use cases can be expanded by additional use cases[1]. Via the *include* relation, these use cases can use other partial use cases[2], whose behavior will be integrated completely.

5.1.4 Activity diagram

The activity diagram belongs to the behavior diagrams. It is particularly well suited to describe the dynamic aspects of a system which is the reason why it is often used to model business processes. Complex workflows can be represented clearly structured, and individual steps of a use case can be modeled graphically. The representation is similar to a web or graph whose nodes represent actions. Flows of control and data are specified at the edges of the graph[3].

The activity diagram shown in Fig. 5.5 describes the change workflow for the product configuration and forms the basis for the prototype implementation (see section 6.2.4).

Activity partitions are used to group the processing steps appropriately. This makes the organizational unit clear which is responsible for specific processing steps.

1 To give examples, the use cases for performing decomposition, editing position, or editing position variants shall be mentioned here.

2 Examples for partial use cases are those of performing workflow or performing appraisal.

3 As of UML 2.0, semantics has distinctly approached Petri nets.

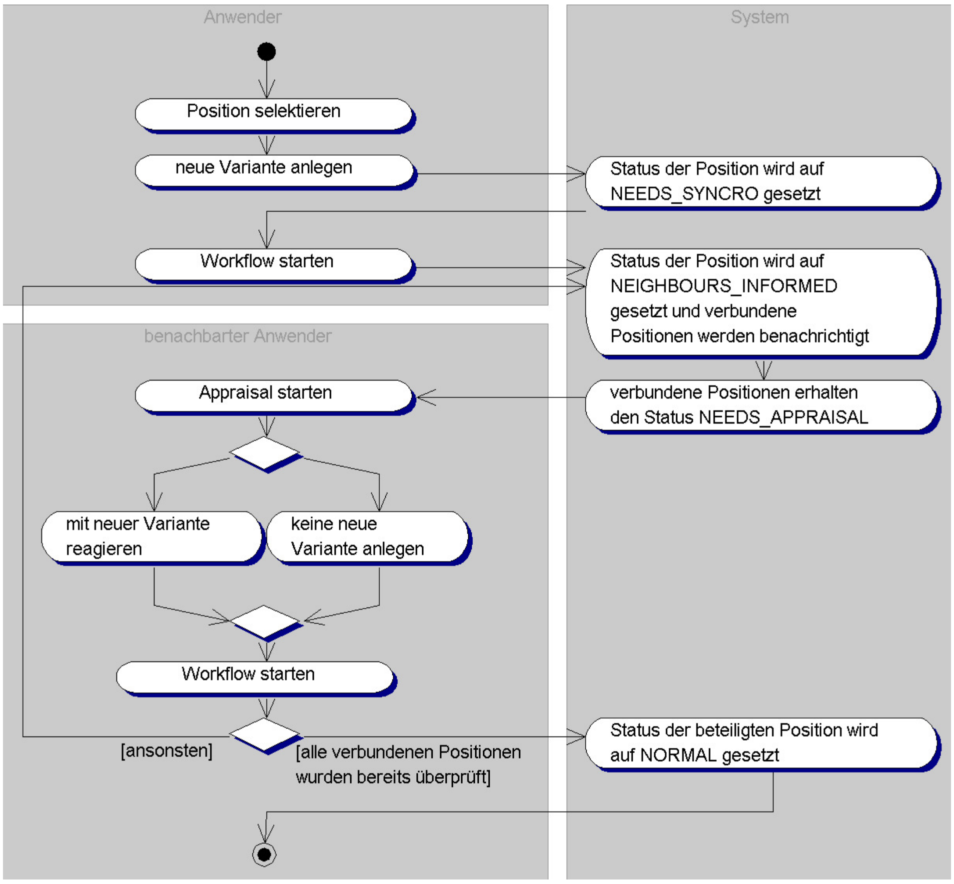


Fig. 5.5: Activity diagram of change workflow for product configuration

The decomposition shown in Fig. 5.6 is another example of an activity diagram for the interconnections documentation.

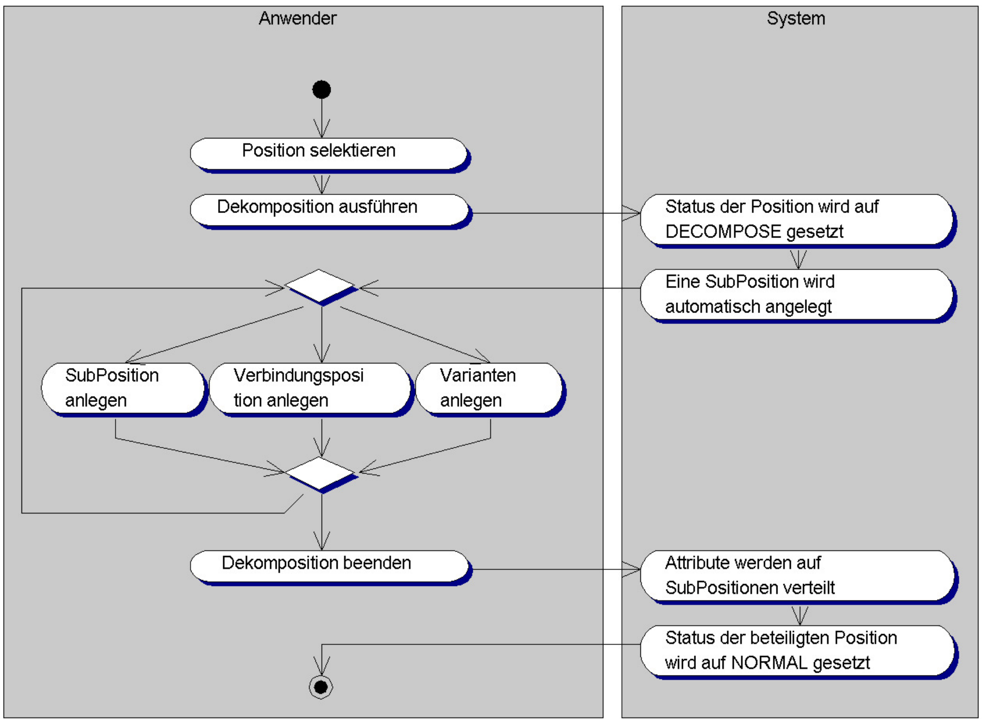


Fig. 5.6: Activity diagram of decomposition

5.1.5 Sequence diagram

The sequence diagram also belongs to the behavior diagrams. It can be used to represent dynamic aspects of a system where the focus is on the exchange of messages between communication partners. The sequence diagram shows the order in which partners communicate with each other to perform a specific task. All participants involved in the interaction are entered in the horizontal axis. The vertical axis defines the order in which the partial tasks are performed.

UML 2 defines the sequence diagram in a very general way talking about participants in the interaction and life lines. Life lines represent time axes. The life lines have activation bars which indicate the area in which the communication partners control the flow and participate actively in interactions. Fig. 5.7 shows the sequence diagram for the decomposition as example.

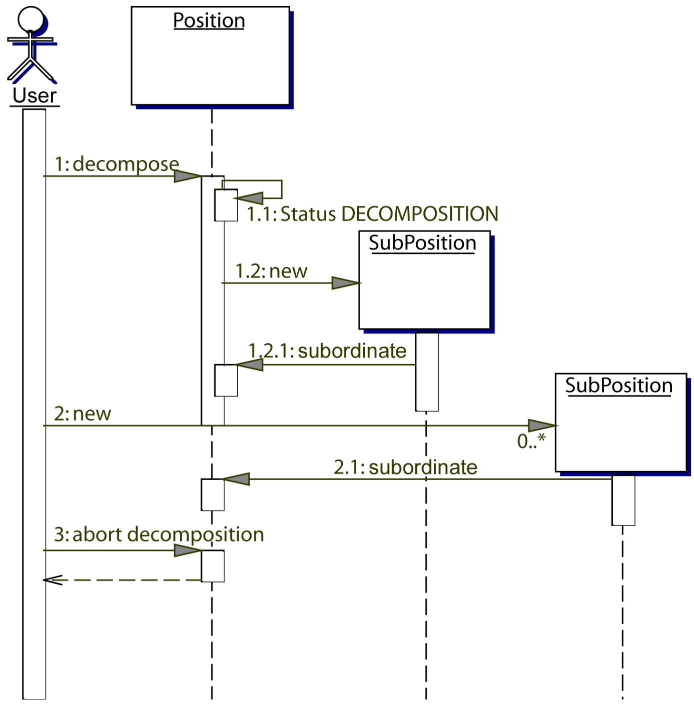


Fig. 5.7: Sequence diagram of decompose

Apart from the order in which the operations are performed, the sequence diagram also describes the respective classes and their objects. In this way, the sequence diagram represents an important link between the features and the class diagram.

5.2 IT architecture of the prototype

The previous section introduced the key components for an object oriented design of the interconnections documentation. Now, the IT architecture relevant for the prototype implementation shall be presented. Section 5.2.1 explains the basic architectural concepts to start with. After detailing the technological basics in section 5.2.2, section 5.2.3 gives an overview of the IT technologies used for the prototype implementation.

5.2.1 Basics of the IT architecture

Different architectural concepts can be distinguished for IT systems. With a two-tier architecture, the GUI and the technical concept are firmly intertwined in one tier. The second tier is used to store the specific data.

In contrast to this, three-tier architectures consist of the GUI tier, the technical concept tier and the data storage tier.

The prototype implementation of the interconnections documentation shall be based on the principles of the three tier architecture which will be detailed further in the following. Fig. 5.8 shows three different tier models the IT systems can be divided into. The MVC model [WiMVC] is less a layer model than rather a pattern for software architecture. It divides software systems into the three units *model*, *view* and *controller*. According to [WiMVC], it is regarded as de-facto standard for a rough design of complex software systems by now.

On the assumption that the view layer can communicate with the model layer only via the controller layer, this slightly modified form also represents a layer model.

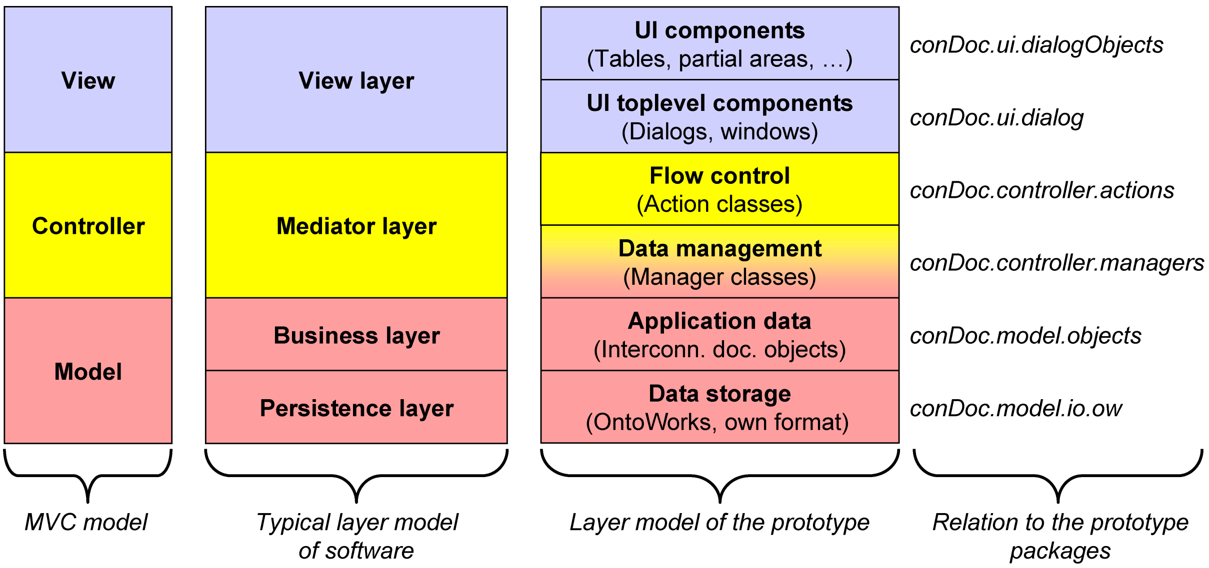


Fig. 5.8: Layer models
(based on [Römer06])

Compared to the MVC model, the layer model in the middle is a typical software layer model. The third (right) layer model is a layer structure with an even finer granularity which is used for the prototype.

In general, the *view* layer contains the part of an application which describes the graphical user interface (GUI), i.e. those classes which are visualized graphically at program runtime and used for user interaction. In the case of the developed prototype, this layer is split into two sublayers.

The *controller* layer contains the actual prototype logic. This layer controls the application. The user interactions with the GUI trigger events which, in turn, trigger defined actions. These actions perform specific tasks. Since the *controller* layer mediates between the view and model layers, it is also referred to as *mediator layer*.

The *model* layer contains the data objects types to be displayed and their relations among each other. This layer cannot create a direct link to the view or controller layer on its own.

Below the model layer, there is the persistence layer which stores the current status of an application to a permanent status. This can be realized in a file or database. This layer can be considered to be independent or interpreted as part of the *model* layer.

Fig. 5.8 only shows a rough overview of the prototype system architecture. The structural, static design of the software can be described in detail using UML class diagrams. Due to reasons of clarity[1], it is favorable to bundle several classes with a similar behavior to *packages* which are described on the *package* level.

5.2.2 Technological basics

This section describes the technology used for implementing the prototype.

The use of the *Java SDK* (*Software Development Kit*) of *Sun Microsystems* [WWWs1] makes the prototype independent of platforms[2]. The implementation in *Java* makes the prototype independent of a specific operating system.

As quasi standard of *Java* developments, *Eclipse*[3] was used as development environment. The GUI of the prototype was mainly designed with *Netbeans* by *Sun Microsystems* since this IDE contains a powerful GUI designer.

By default, the prototype stores the collected data of the interconnections documentation in an own proprietary file format. In addition, there is the possibility to export the entered data to *Protégé*[4] and *OntoWorks*.

Protégé is a program of Stanford University developed in Java which allows the modeling and representation of knowledge based on classes, hierarchies or relations thus providing the possibility to create ontologies. *OntoWorks* is an enhancement of *Protégé* developed by Daimler AG which has been adjusted to corporate specific requirements.

---

1 The prototype consists of 220 classes.

2 In EDP, the term "platform" has different meanings or degrees of abstraction (cf. glossary).

3 *Eclipse* is a freely available development environment (IDE) by *Sun Microsystems*.

4 Protégé [WWWp1] is developed by *Stanford Medical Informatics* at *Stanford University School of Medicine*.

The prototype implementation mainly uses the *OntoWorks* API which expands the *Protégé* API. To realize certain functions, like e.g. exporting, the *Protégé* API is addressed directly. For the course of this study, differentiation between *Protégé* and *OntoWorks* will not be made since it does not have any effects on the interconnections documentation.

To store the data of the interconnections documentation as *Protégé* project, a meta model of the data has to be created at first. This meta model is loaded via the *Protégé* API. It is only then that the application data can be exported to this structure. The meta model was specified as UML class diagram with *Rational Rose* by *Rational Software* and converted into a *Protégé* meta model using a plug-in.

5.2.3 Overview of the used technologies

The technologies used can be grouped as follows:

- Programming language: *Java SDK 1.4* by *Sun Microsystems*
- Development environments: *Eclipse 3.1* by *IBM*; *Netbeans 3.6* by *Sun*
- Used programming interfaces: *Protégé* API version 2.0.1, *OntoWorks* API version 1.1.2
- Export compatible to file format: *Protégé-2000*, *OntoWorks* Version 1.7
- *Protégé meta model created in:* *Rational Rose by Rational Software*

5.3 Realization of the prototype

Based in the architectural concept described in section 5.2, this part of the study deals with the details of the IT realization. Section 5.3.1 presents the layer model in its entity the prototype realization is based on. The different layers (presentation, control, application and persistence layer) are described separately in sections 5.3.2 to 5.3.5.

5.3.1 Layer model of the prototype

Fig. 5.9 shows the most important classes and class groups of the prototype, their mutual associativity, their assignment to the MVC model, and their allocation to *packages*.

The classes in Fig. 5.9 do not represent individual classes, but rather groups of classes. For a detailed representation, the reader is referred to the appendix. In the following, the various layers shall be described individually.

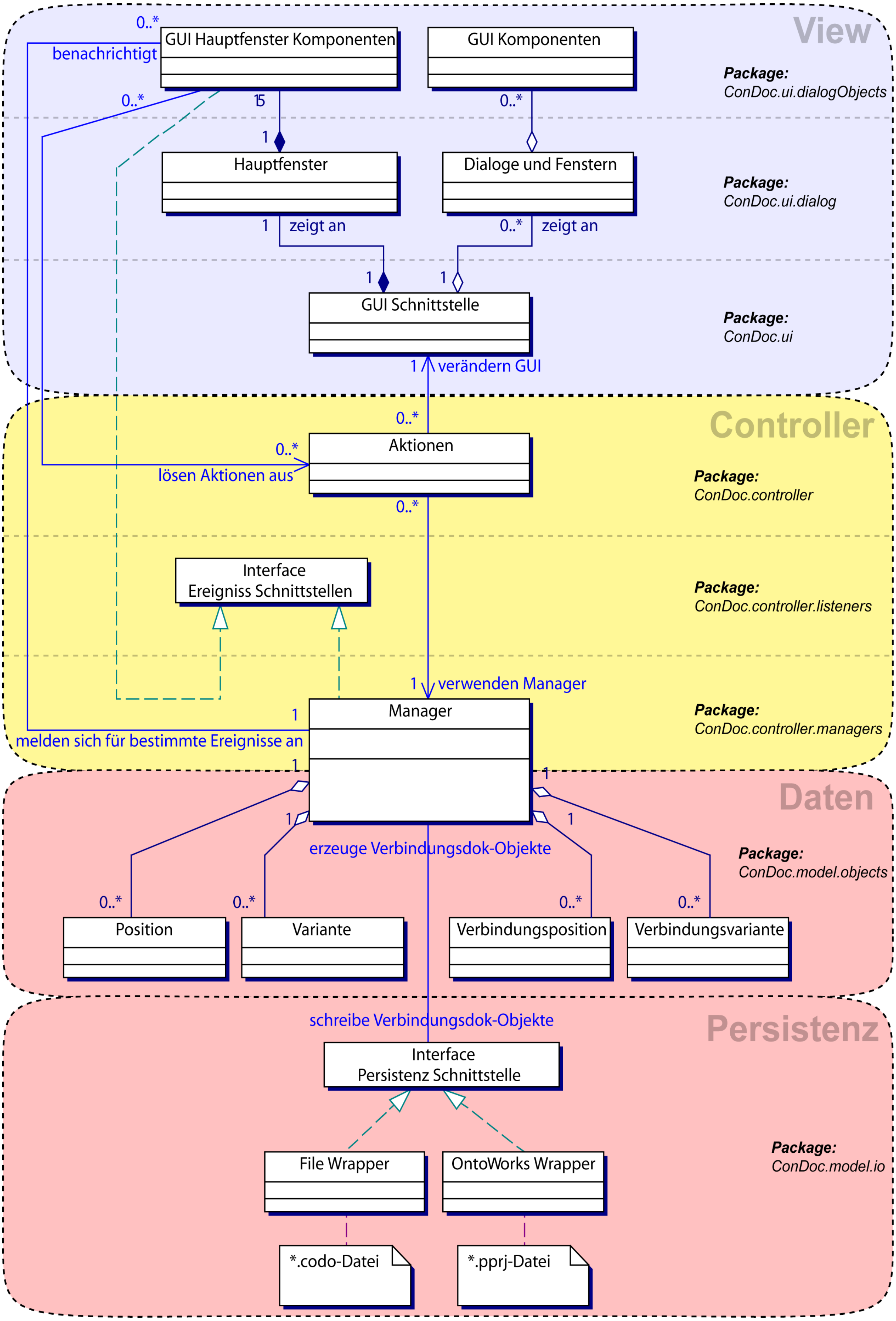


Fig. 5.9: Hybrid representation of the system architecture as layer model and class diagram

5.3.2 View layer

The view layer comprises all classes of the demonstrator which can be displayed graphically. They are in the package "conDoc.ui". The dialogs of the application are stored in the package "conDoc.ui.dialog". These dialogs are composed of the GUI components stored in the package "conDoc.ui.dialogObjects".

The GUI components represent portions of the dialogs and are specializations of the Java class "javax.swing.JPanel". The GUI components themselves are built with Java components which belong to the Java language.

5.3.3 The controller layer

As shown in Fig. 5.9, the GUI components communicate directly with the *model* layer. And vice versa, the *model* layer does not know the *view* layer. To be precise, this is not in line with the MVC idea, but offers a few advantages:

- The architecture of the demonstrator can be represented as layer architecture.
- The communication between *view* and *model* must be via the controller layer which, thus, has the complete control over all flows.
- The interconnections documentation objects remain free (thus lean) of unnecessary methods which have nothing to do with the current system status.

5.3.4 Application data layer

The application data layer contains 24 data object types[1] whose instances represent the current status of the application. As already described in section , these objects cannot access the view or controller layer. The instances of these classes are managed by specialized managers of the controller layer.

Fig. 5.10 shows those object types which are absolutely mandatory to map the interconnections documentation.

The abstract superior class "CObject" itself is not instantiated, but simply defines the general structure and the generally valid behavior of the derived objects "CPosition", "CVariant", "CConnection" and "CConnectionVariant". All classes manage any number of attributes.

1 They are in the *package* "conDoc.model.objects" or the *subpackages*.

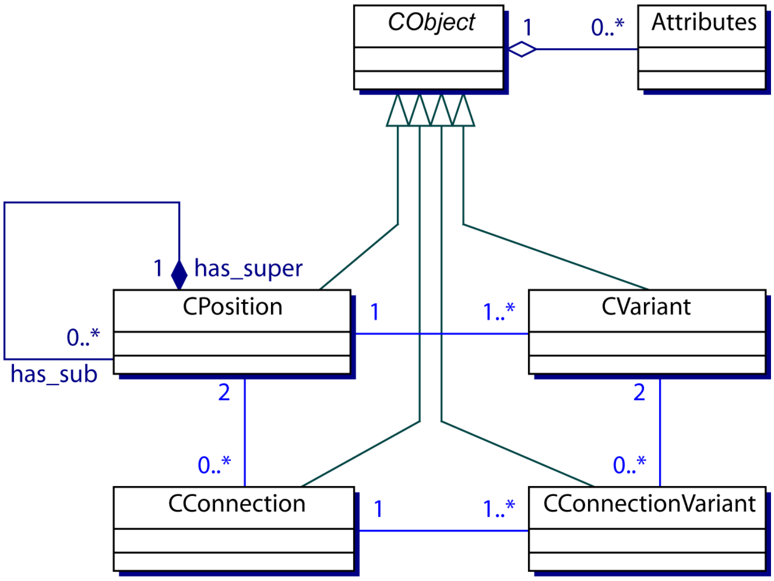


Fig. 5.10: The application data structure

5.3.5 The persistence layer

The persistence or data securing layer of the prototype files the current status of the application to a permanent status. Typically, this is realized in the prototype specific file format.

In addition, there is also the possibility to export the data of the interconnections documentation to *Protégé* and *OntoWorks* (cf. Section 5.2).

When loading a project, the serialized data are used to generate instantiated objects again. The objects obtained this way possess the same characteristics as the original application objects.

In the context of the interconnections documentation, a defined meta model is used as foundation which is defined at the beginning and remains unchanged. The meta model also defines the structure the data are stored in.

The *Protégé* API stores the exported project in three files. Compared to the proprietary prototype format, these files are very large since they have an XML similar structure. Loading time is very long for extensive projects. Therefore, the internal format shall be used preferably.

6 Prototype implementation

This chapter presents the prototype implementation of the interconnections documentation. At first, the general structure of the prototype and its GUI are explained in section 6.1. The key functions to handle the demonstrator will be introduced. Using the business scenario "ITego" as example, section 6.2 describes the step-by-step documentation of the interconnections along the product generation process and the way the prototype works in detail.

6.1 Basic GUI structure

Fig. 6.1 shows the principle structure of the demonstrator GUI. Apart from the menu bar, the GUI is split into four main areas:

- **Conventional top-down structure:** This area is in the top left section of the window (cf. Fig. 6.1 / 1) and represents the product as hierarchical tree. The representation is in top-down from the entire product to the SDEs. By "expanding" the structuring nodes, the product can be run through from top to bottom.
- **Graphical web structure:** This part of the GUI is in the top right section of the window (cf. Fig. 6.1 / 2). For the loaded product structure, the tabs **Product structure**, **Intercon. structure** and **User views** can be selected for a graphical representation of the product structure (cf. section 6.1.2).
- **Functional views:** The bottom left section of the window shows the functional views of the product (cf. Fig. 6.1 / 3). The user can select between a hierarchical representation of the functional views and a list of the views.
- **Product and process information:** This part of the GUI is in the bottom right section of the window (cf. Fig. 6.1 / 4). There is a total of nine tabs which offer different possibilities of managing product and process specific detail information on the products or customer orders. The information is presented in the form of lists or tables.

The different areas are described in detail in sections 6.1.1 to 6.1.4.

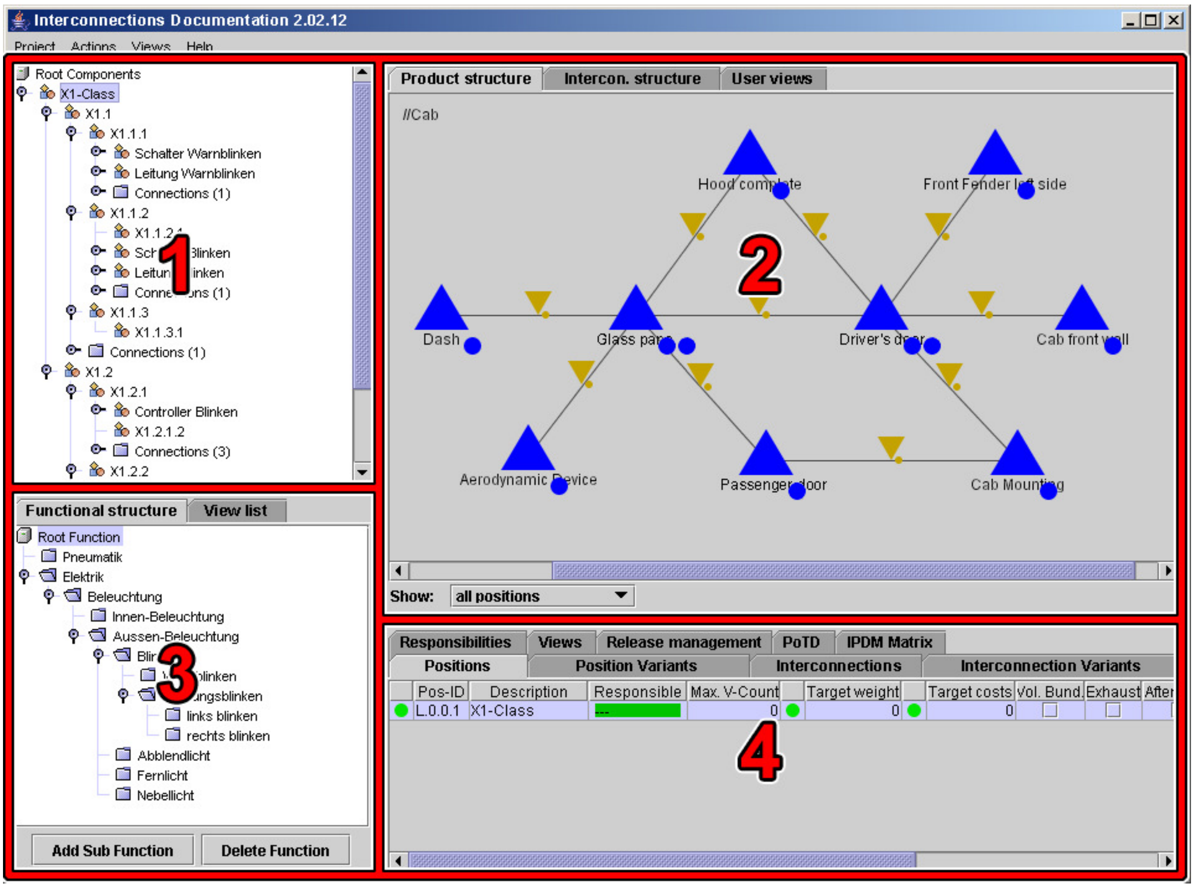


Fig. 6.1: Basic GUI structure

The menu bar (cf. Fig 6.1) contains the four menu items:

- **Project**
- **Actions**
- **Views**
- **Help**

The menu items which are realized as pull-down menus are used to open the key functions[1] of the prototype. The context menu[2] offers the user a variety of other commands which depend on the selected GUI objects and their current status. This type of implementation keeps the application GUI clearly structured and minimizes wrong user entries.

6.1.1 Conventional top-down structure

The tree view component offers a graphical representation of the product structure which allows the user to obtain a fast overview of the product. The navigation is as in customary structure browsers. The product is structured hierarchically from top to bottom. With this type

1 The individual functions are not described here since this would go beyond the scope of this study.

2 It is activated via the right mouse button.

of representation, the product is represented by part positions, part position variants and interconnection positions[1] (cf. Fig. 6.2).

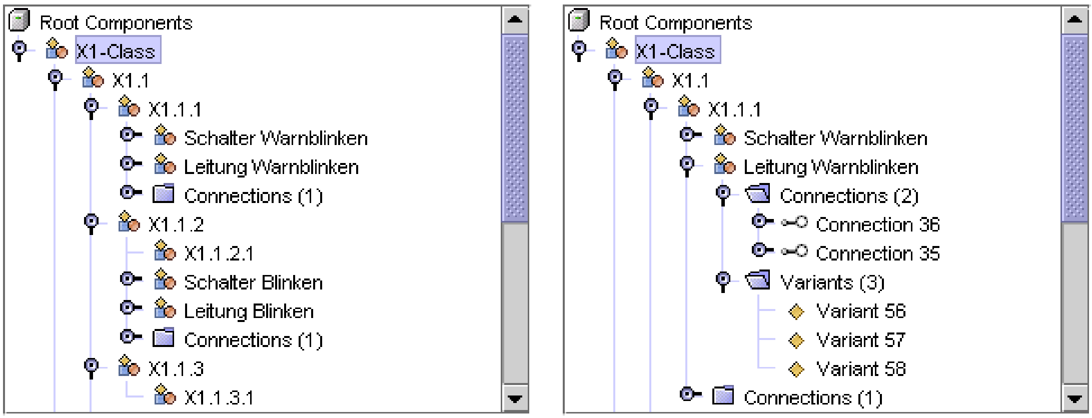


Fig. 6.2: The conventional top-down product structure

The structuring can comprise any number of levels. Part positions are composed of their subpositions and can have any number of interconnections. The part position variants are at the bottom level of the product structure. New structural elements can be created either in the main menu via the pull-down menu **Actions** or the context menu.

6.1.2 Graphical web structure

Typically, the graphical representation of the product as web of part and interconnection positions requires the largest space on the screen.

The **Product structure** tab displays the product as hierarchical product structure as in a tree view described in section . The representation of variants is omitted here to give space to other information (e.g. the engineering responsibility for a defined product component). Selected positions can be displayed in detail and edited in the tabs **Positions** and **Interconnections** (cf. section 6.1.4) via the context menu. They can also be added to a view.

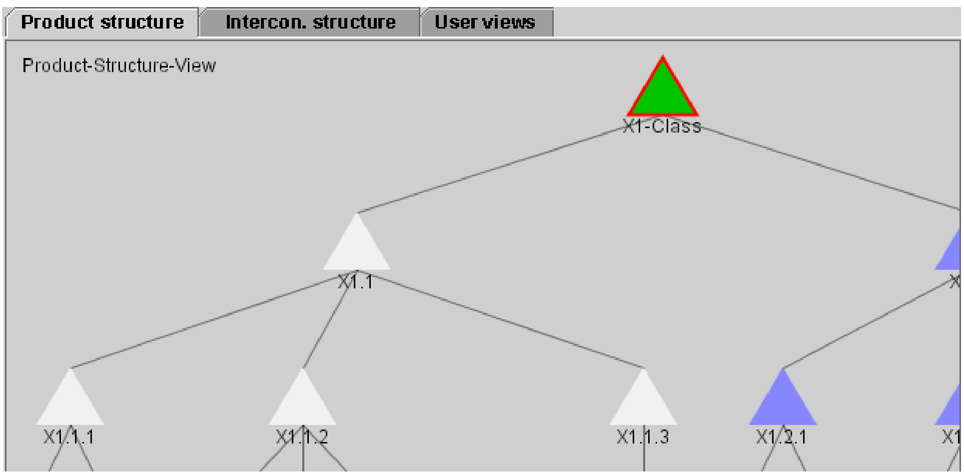


Fig. 6.3: The hierarchical product structure as graphic representation

1 Interconnections are not visualized in this representation.

The second tab shows the product structure as web of part and interconnection positions (cf. Fig. 6.4).

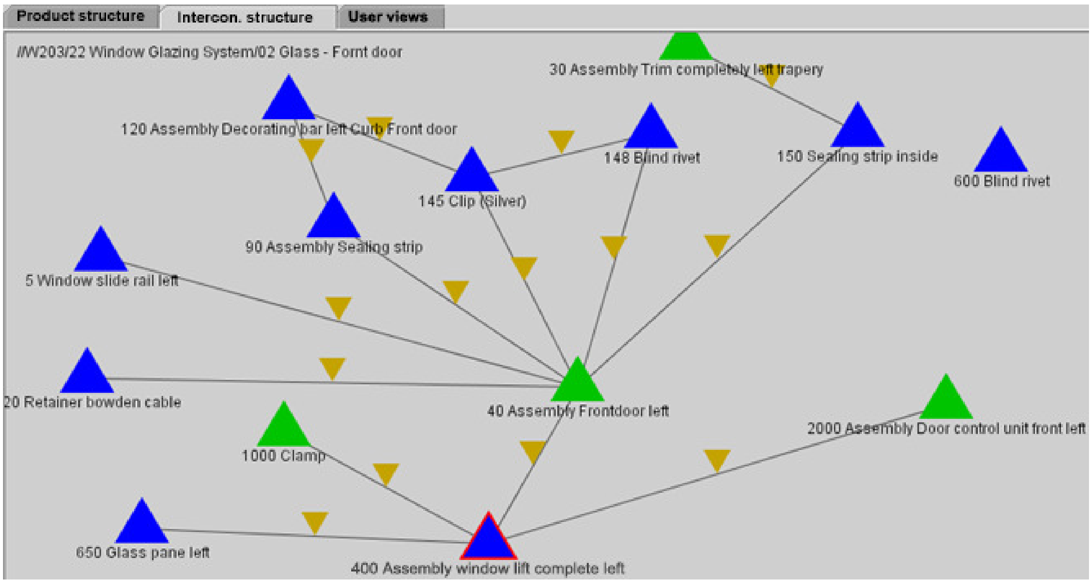


Fig. 6.4: The flat product structure as graphic representation consisting of part and interconnection positions

As for the display in the **Product structure** tab, detail information for selected positions can be visualized in the tabs **Positions** and **Interconnections** (cf. section 6.1.4). Based on this representation, the properties of the positions and variants can be edited and their status can be changed. The workflows **Appraisal** and **Decomposition** implemented for the prototype, and the **Bottom-Up Wizard** are also started from here (cf. sections 6.2.4 and 6.2.6).

6.1.3 Functional views

The **Functional structure** tab realizes the concept of process specific views in the prototype as described in section 4.8. In contrast to the product structure, the **Functional structure** tab is a taxonomy.

The **Functional structure** tab is defined by the bundling of part positions. A functional structure or view is based on a group of defined positions. Views which are defined exclusively by bundling positions are called **PRIMARY**. Views can also be combined among each other to build hierarchies. Hierarchical views belong to the category **SECONDARY**.

As for the product structure, a functional view is displayed as tree view component in the prototype. Fig. 6.5 shows an example of such a functional, hierarchical view.

An overview of all defined views of a product can be displayed in the **View list** tab (cf. Fig. 6.6).

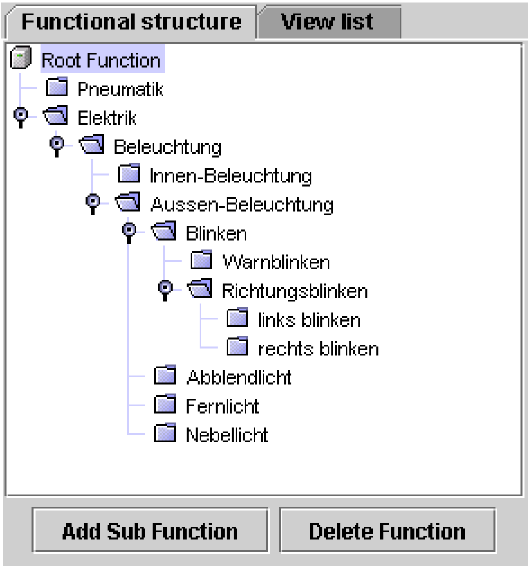


Fig. 6.5: **Functional structure** tab

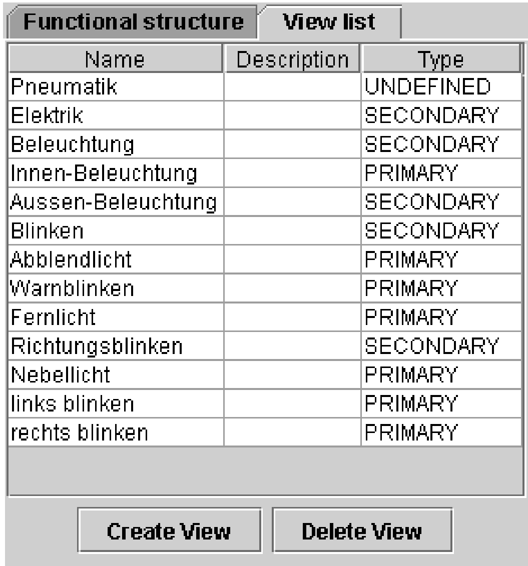


Fig. 6.6: **View list** tab

6.1.4 Product and process information

The bottom right section of the GUI offers nine tabs for detailed product and process information. With the concise representation of the information as table, it is possible to display the user many details on the relevant interconnections documentation objects. Fig. 6.7 shows a screenshot of the nine tabs.

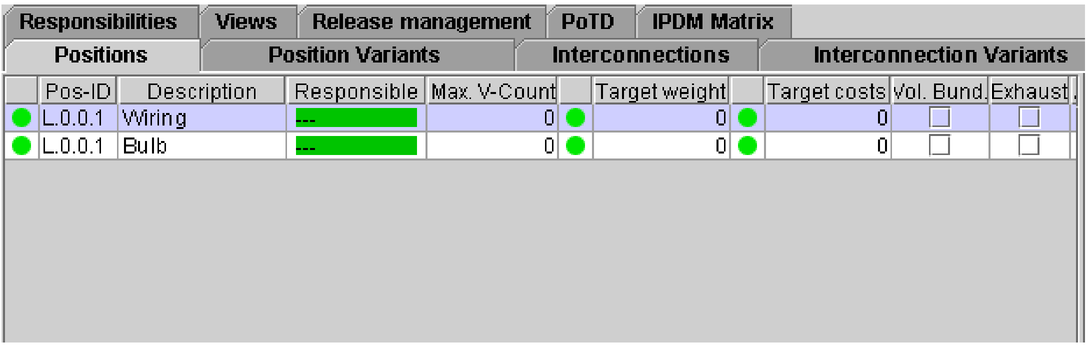


Fig. 6.7: Product and process specific detail information

The **Positions** tab shows all part positions (including their attributes) which are on the same hierarchical level. The **Position Variants**, **Interconnections** and **Interconnection Variants** tabs show the same behavior as the **Positions** tab (for their specific information).

The **Responsibilities** tab shows the responsibilities of the involved organizations based on positions. In relation to the selected organization, only those positions are displayed which match the selected responsibility.

The **Views** tab shows an overview of the existing customer orders. For a selected order, the vehicle configuration is displayed as code bar.

The **Release management** tab shows all part position variants of all hierarchical levels that exist for a defined maturity level.

The **PoTD** tab displays all interconnections documentation objects that are in direct relation to a defined position.

Integrating the product generation process and customer order process in the **IPDM Matrix** tab means the realization of an essential component of the interconnections documentation. This feature can be used to work on complex tasks:

- Customer orders can be configured, changed or deleted individually.
- The manufacturing sequence of customer orders can be changed (when feasibility restrictions are infringed e.g.).
- Parts demand calculation can be performed for defined customer orders.
- Workplans can be created individually for each vehicle defined as customer order (bottom-up).
- The variants to be used for an order and their code rules can be determined.
- Stock evaluations (with allowance for the calculated part demand) can be performed.

6.2 Business scenario "ITego"

Section 6.1 focused on describing individual system functions, while this section describes a continuous business scenario.

6.2.1 Scope and data basis

The following sections present the prototype implementation of the interconnections documentation using the business scenario "ITego" as basis.

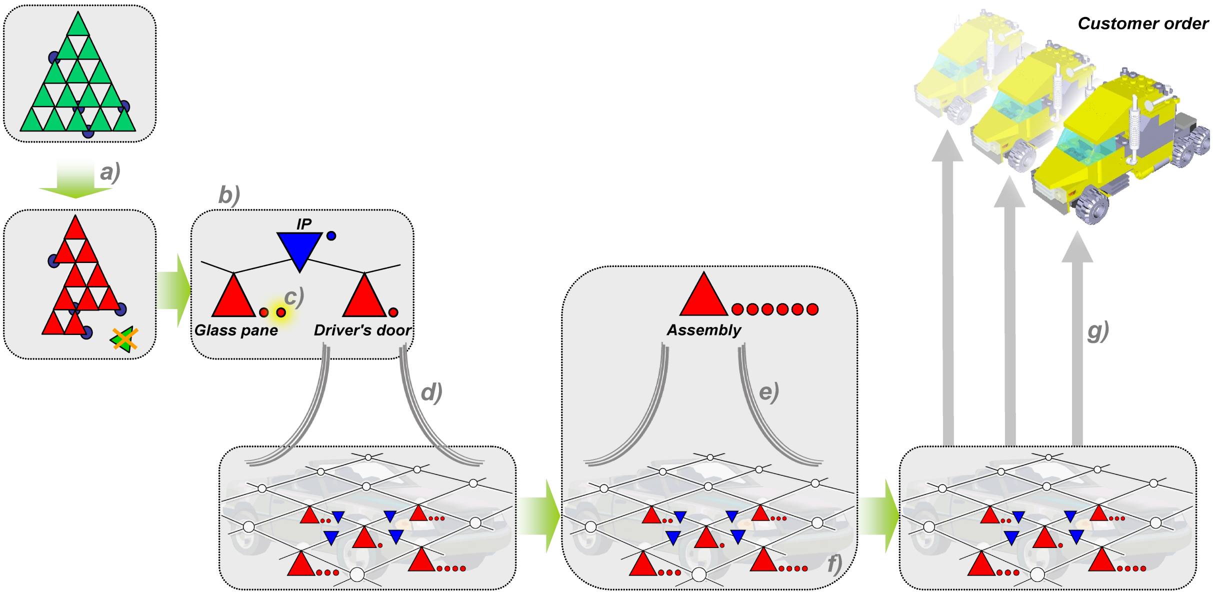


Fig. 6.8: The business scenario "ITego"

According to the process shown in Fig. 6.8 a) to g), the following seven steps are performed:

- derive the psPS from the OPS (see section 6.2.2)
- document the product/process information (see section 6.2.3)
- configure an additional (pane) variant (see section 6.2.4)
- decompose the driver's door (see section 6.2.5)
- bottom-up generation (see section 6.2.6)
- append attachments (see section 6.2.7)
- apply the IPDM matrix (see section 6.2.8)

A LEGO car is used as data basis for the scenario presented in this study. In the following, the model car is referred to as "ITego" which serves the prototype as database. Without variants, the "ITego" has the following properties in the initial version:

- Number of parts: 52
- OPS elements[1]: 51
- Part positions: 97
- Part position variants: 97
- Interconnection positions: 122
- Interconnection position variants: 122

When modeling the product and process information of the "ITego", the feature information was considered as SDEs.

1 Vehicle, Navigation Systems, Navigation Groups, Functional Modules.

6.2.2 Creating the project specific product structure

To derive the project specific product structure from the OPS, a simplified method was used for the prototype implementation (compared to the concept presented in section 4.2).

When the OPS copy is loaded (cf. Fig. 6.9 a), the positions which are not relevant for the psPS can be deleted from the product schema (cf. Fig. 6.9 b). Deleting a position node deletes all structuring nodes which are subordinate to this position. The product structure created this way can be stored as psPS (cf. Fig. 6.9 c) to be available to the further process.

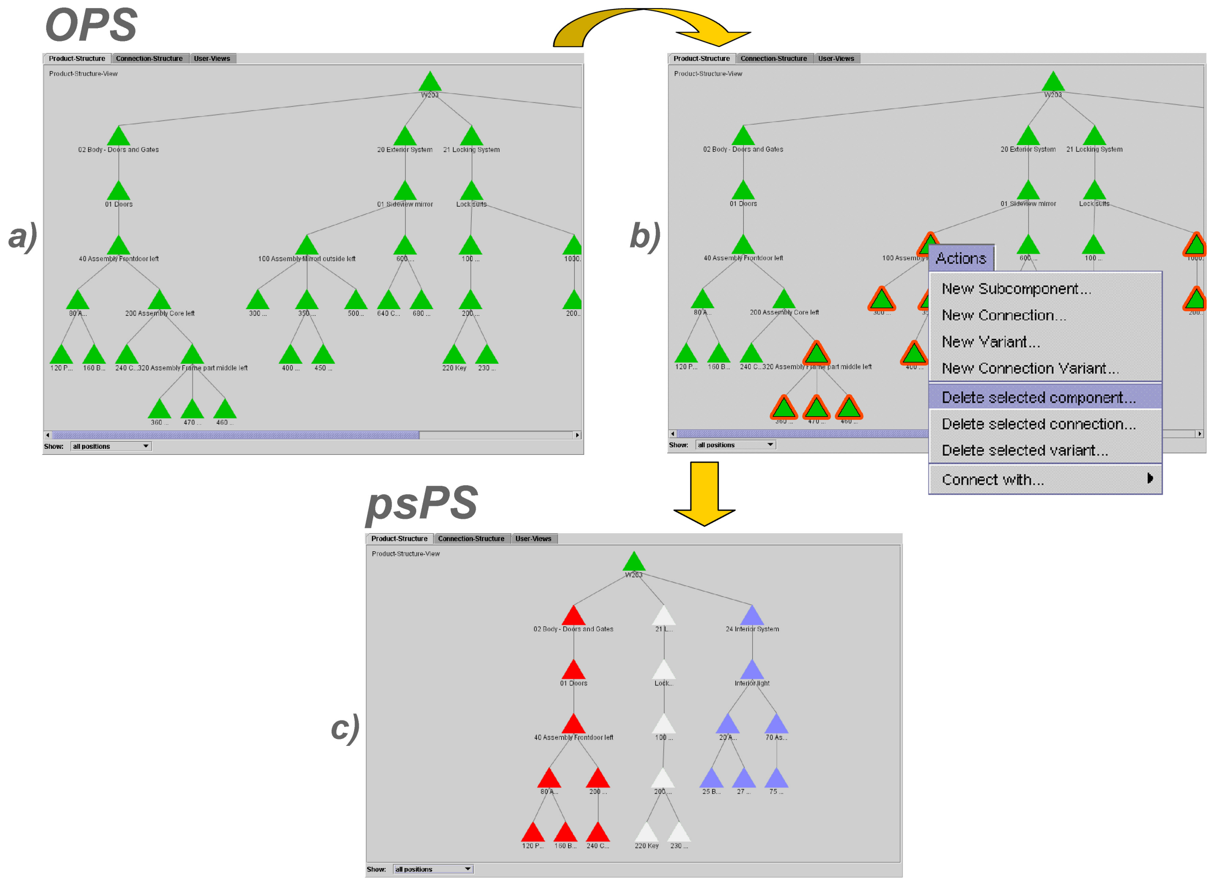


Fig. 6.9: Deriving the psPS by deleting positions from the OPS

6.2.3 Documentation of product and process information

After deriving the psPS from the OPS, the information for the "ITego" comprises the partial web shown in Fig. 6.10. For all part and interconnection positions, the respective basic version variants[1] exist at that point in time. For the scenario of the "ITego", the interconnection **Glass pane/Driver's door** and its part positions **Driver's door** and **Glass pane** will be reviewed in detail in the following.

1 The basic version variant represents the status "series" (cf. section 4.3).

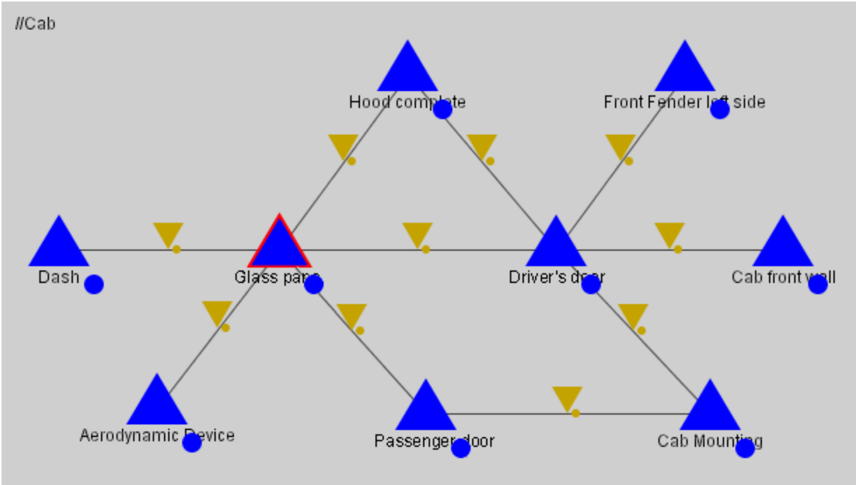


Fig. 6.10: Partial web of "ITego"
(Glass pane and Driver's door)

To edit the position **Glass pane**, the dialog box shown in Fig. 6.11 can be opened via the context menu or double clicking. Due to reasons of clarity, the dialog box is divided into different areas[1]. In the following, the different data areas are introduced. For the detailed "ITego" information, the reader is referred to the respective prototype screenshots.

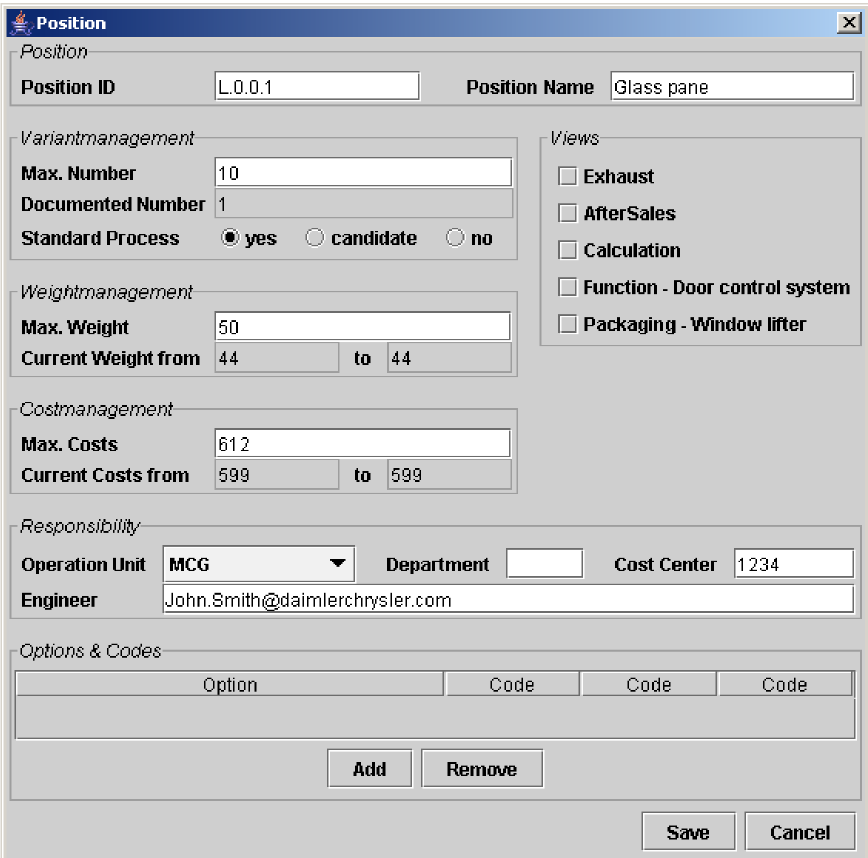


Fig. 6.11: Part position **Glass pane**

1 The layout of the different dialog boxes to separate part positions and interconnection positions and their variants is uniform for all dialog boxes. The different areas will be explained using the position **Glass pane** as example. For the subsequent dialog boxes, the description will be limited to deviations.

The data area **Position** shows the ID number and position name. Information in the **Variantmanagement** area is used to control and manage the part and product variants. The attribute **Max. Number** is used to define the maximum number of variants for the position **Glass pane**[1]. As for the variant management, the areas **Weightmanagement** and **Costmanagement** define the maximum weight and maximum costs for a pane[2]. The area **Responsibility** is used to define detailed information regarding development responsibility. The codes that affect positions are stored in the area **Options & Codes**. Based on these codes, the code rules of the related part position variants are determined computer-aided (cf. Fig. 6.12). The part position is assigned to one of the predefined views in the area **Views**.

When the editing of the position relevant data is completed, the dialog box for the basic version variant **Glass pane standard** shown in Fig. 6.12 is opened via the context menu or a double click.

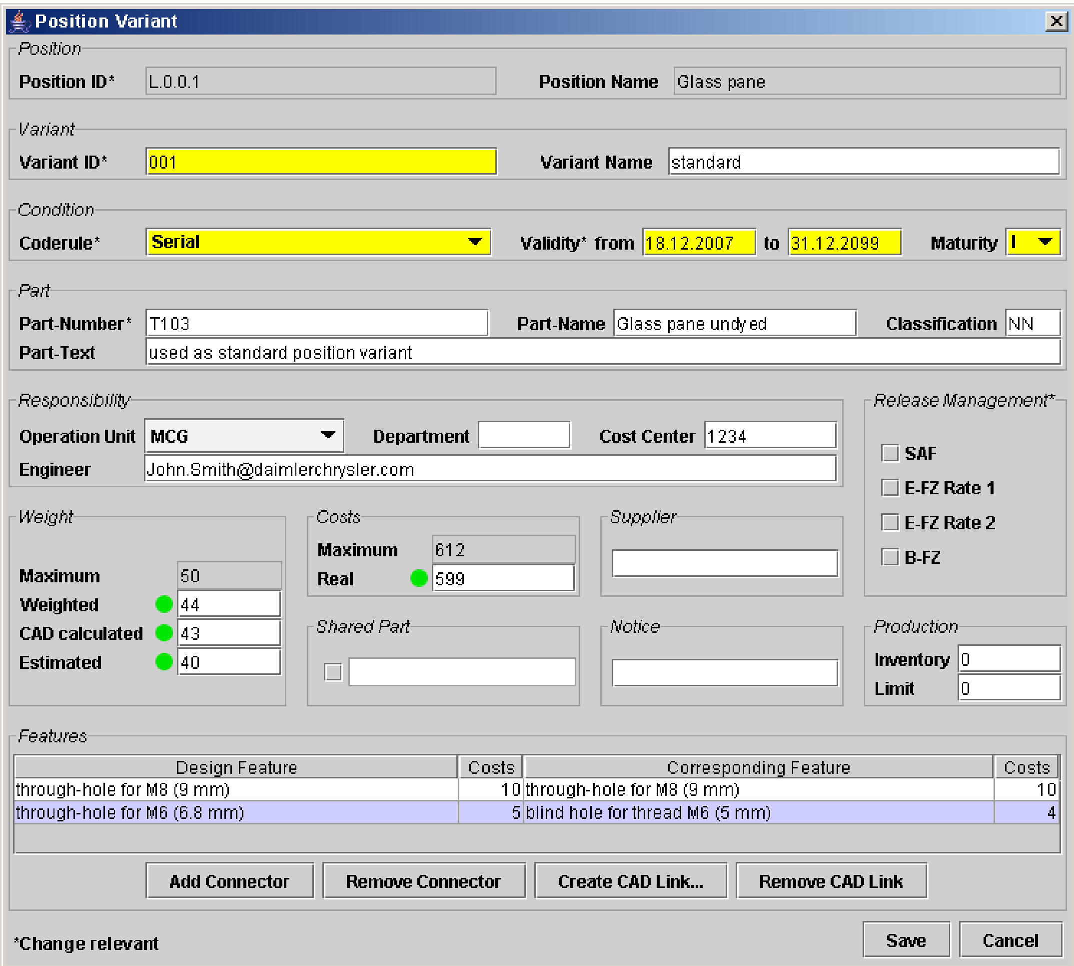


Fig. 6.12: Part position variant **Glass pane undyed**
Basic version variant (= status "series")

1 The data field **Documented Number** shows the number of the actually documented variants on position level.

2 The data field **Current Weight from** shows the minimum or maximum weight for the actually documented variants on the position level. The same applies for all other **Current ... from** fields (cf. costs, times, etc.).

While the selection condition (= code rule) for the part position variant is defined in the area **Condition**[1], part relevant information is documented in the data area **Part**. **Responsibility** can be used to differentiate defined responsibility on the position side on variant level. The fields in the areas **Weight** and **Costs** are used to document weight and cost information. A stoplight function is used to visualize whether the specifications on the position side are met. In the **Features** area, the **Add Connector** button can be used to define the outside geometries of the parts. By selecting the design features, possible corresponding features for the part position variants to be connected are defined. The data area **Supplier** is used to document supplier data. The **Shared Part** area forms the basis for the commonality management. Stock information can be documented in the area **Production**. Part position variants are related to the release process via the **Release Management**. The area **Notice** is used for general remarks.

In the dialog box **Glass pane/Driver's door**, the interconnection between the part positions **Glass pane** and **Driver's door** can be edited (cf. Fig. 6.13).

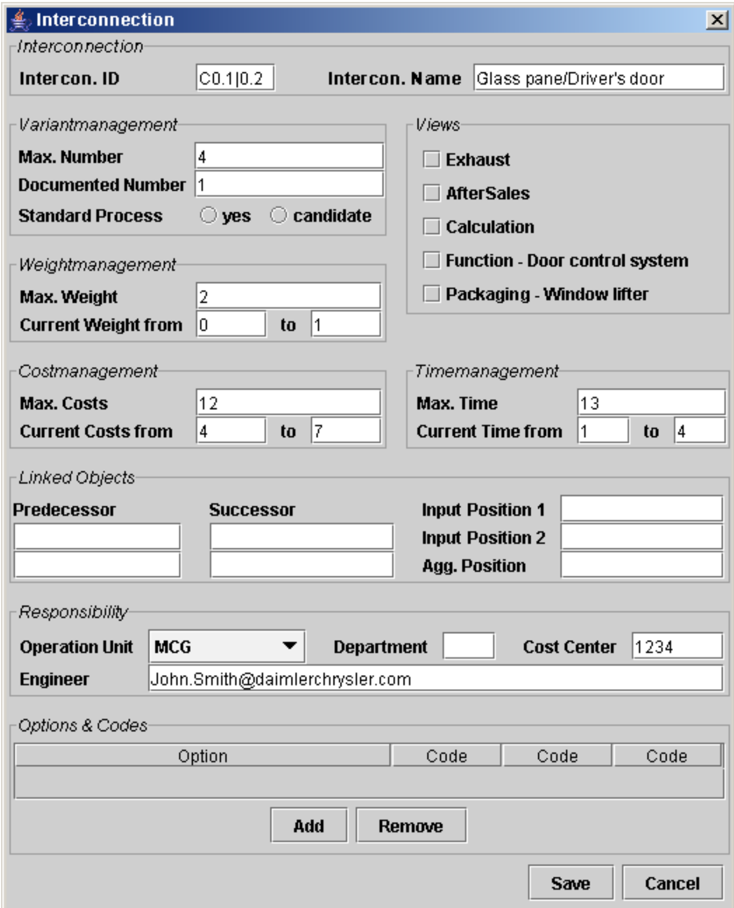


Fig. 6.13: Interconnection position **Glass pane/Driver's door**

1 The selection conditions are generated from the position affecting codes (see dialog box of part position). They are available for selection at the part position variants. A code rule selected for a defined variant is no longer available for selection for another variant of this position. Code rules which are generated by the system and which are not used actively for a variant by the position responsible user are documented by default as non-feasible variants at the part position. This is to support a complete documentation on the system side.

In the area **Linked Objects**, preceding and succeeding interconnections of the interconnection position **Glass pane/Driver's door** are managed. The fields **Input Position** show the part positions which are linked via this interconnection. Possible assembly positions are shown in the data field **Agg. Position**. The **Timemanagement** is used to define valid maximum assembling times.

Fig. 6.14 shows the dialog box of the interconnection position variants of the basic version **Glass pane/Driver's door**.

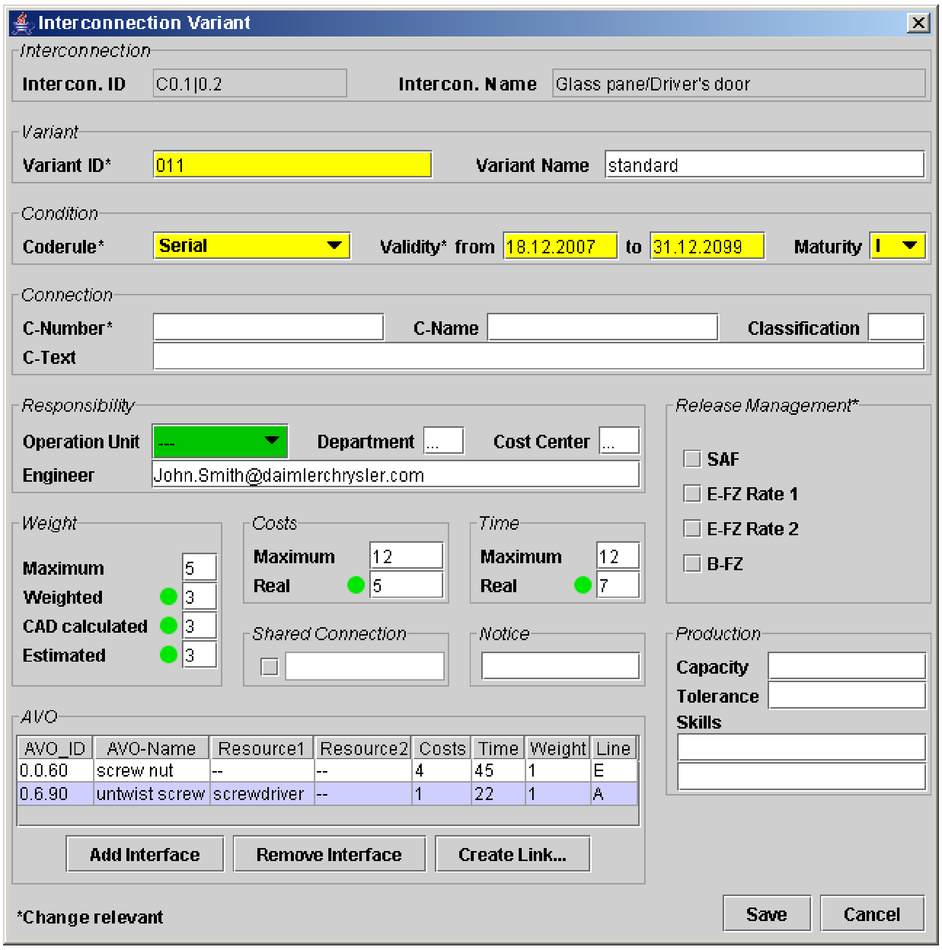


Fig. 6.14: Interconnection position variant **Glass pane/Driver's door** as basic variant

By selecting an interface (**Add Interface** button) in the **AVO** area (operations), the work steps stored for this interface are added to the position variant[1]. The data area **Production** manages production relevant information.

1 Depending on the selection of a design/corresponding feature, the related interface will be selected by the system.

6.2.4 Configuring an additional pane variant

Apart from the (transparent) pane used in series so far, another (blue toned) pane shall be offered. In the following, the additional part position variant **Glass pane blue toned** which has to be considered will be introduced based on the workflow feature of the prototype[1].

At first, **C1** is documented for the part position **Glass pane** as additional (position affecting) code (cf. section 6.2.3, Fig. 6.11). The context menu of the position **Glass pane** is used to create a new part position variant via the function **new variant**. These part position variant **Glass pane blue toned** with code rule **C1** are edited in the dialog box shown in section 6.2.3 (cf. Fig. 6.12). When the attributes of the variant are stored and the dialog box is closed, the graphical view of the interconnections structure shows the partial web shown in Fig. 6.15. Now, the position **Glass pane** is in the status **Needs synchronization** (visualized by a brown coloring). All positions that have an interconnection to the position **Glass pane** might have to be adjusted.

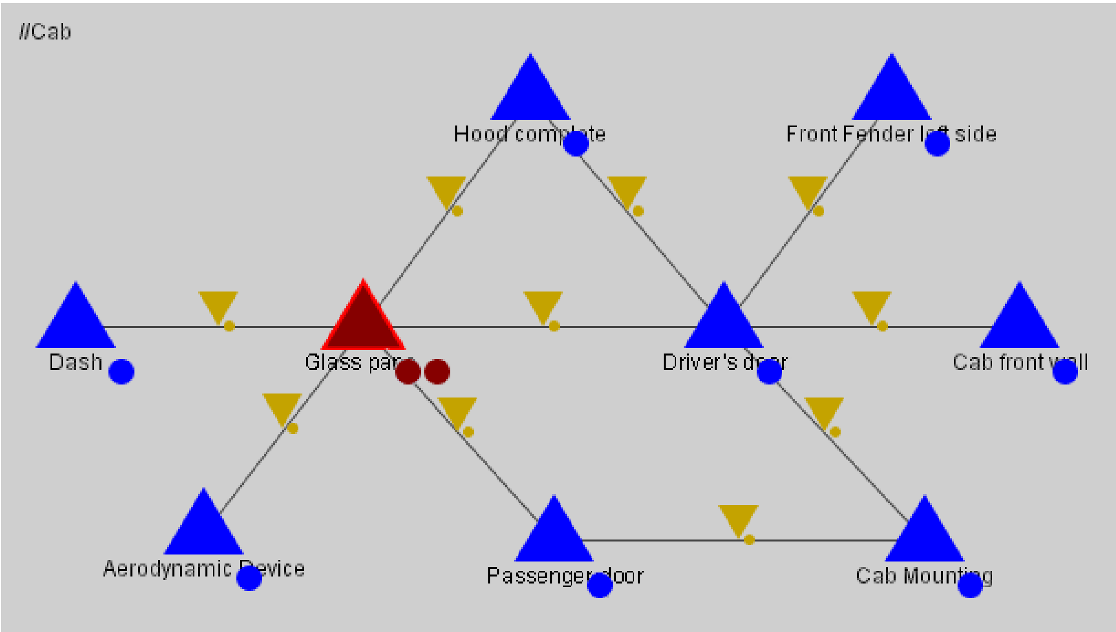


Fig. 6.15: Position **Glass pane** in the status **Needs synchronization**

To start synchronization with the connected positions, the command **Start workflow** is opened via the context menu of the position **Glass pane**.

Glass pane will enter the status **Neighbors informed** (visualized by a green coloring). All neighboring positions connected to **Glass pane** will change to the status **Needs appraisal** as shown in Fig. 6.16. As an example for all affected part positions, the position **Driver's door** shall be investigated in the following.

1 The workflow implementation is based on the configuration concept described in section 4.3.5.

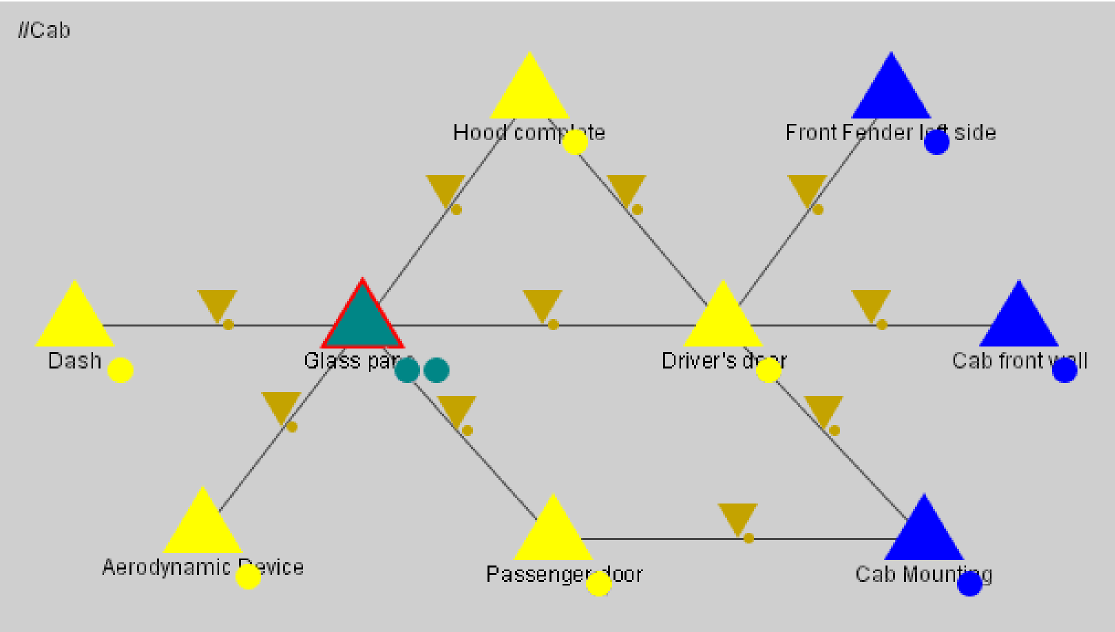


Fig. 6.16: Position **Driver's door** in the status **Needs appraisal**

To evaluate the effects on the position **Driver's door**, the command **Start appraisal** can be opened via the context menu. As shown in the dialog box of Fig. 6.17, the **Create new variant** button is used to create a new position variant with code rule **C1**[1].

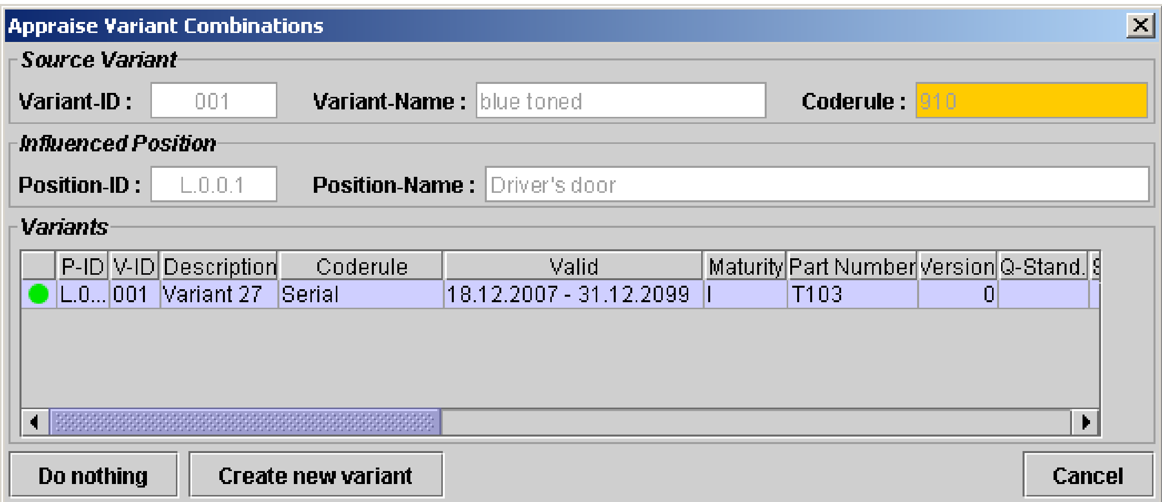


Fig. 6.17: Dialog box to evaluate **Driver's door**

As shown in Fig. 6.18, there are two variants for the position **Driver's door** now.

1 As an alternative, you can press the **Do nothing** button. In this case, no additional variant will be created.

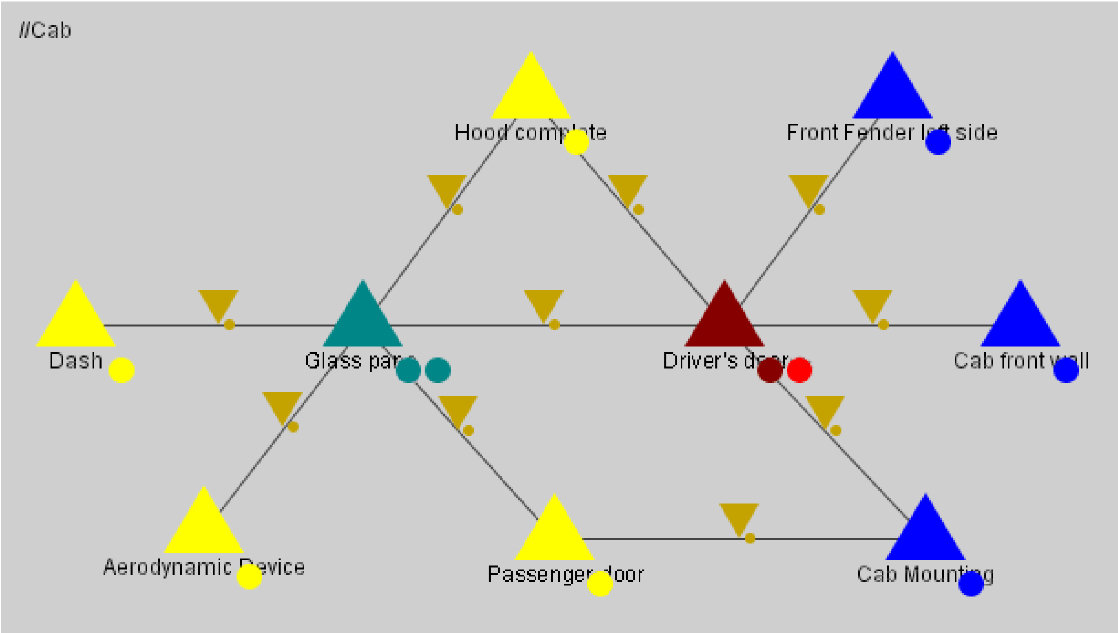


Fig. 6.18: Partial web of "ITego" after creating a new variant for **Driver's door**

To complete the workflow, the effects on the interconnection **Glass pane/Driver's door** have to be evaluated (cf. Fig. 6.19).

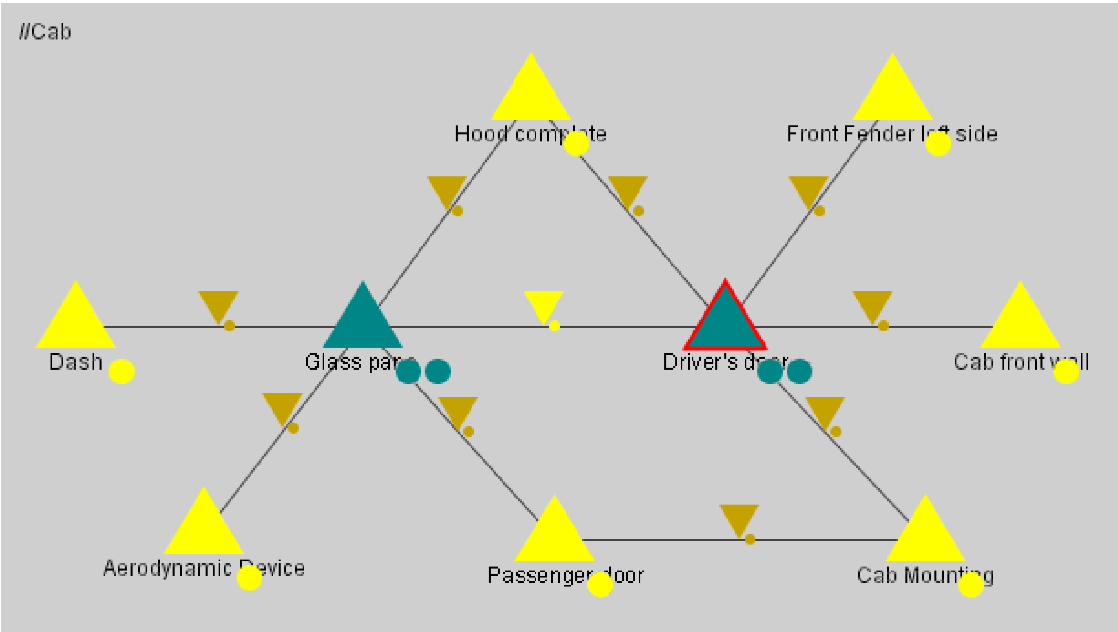


Fig. 6.19: Partial web of "ITego" after evaluating **Driver's door**

Now, the entire change workflow is completed, and all involved positions receive their normal status (see Fig. 6.20).

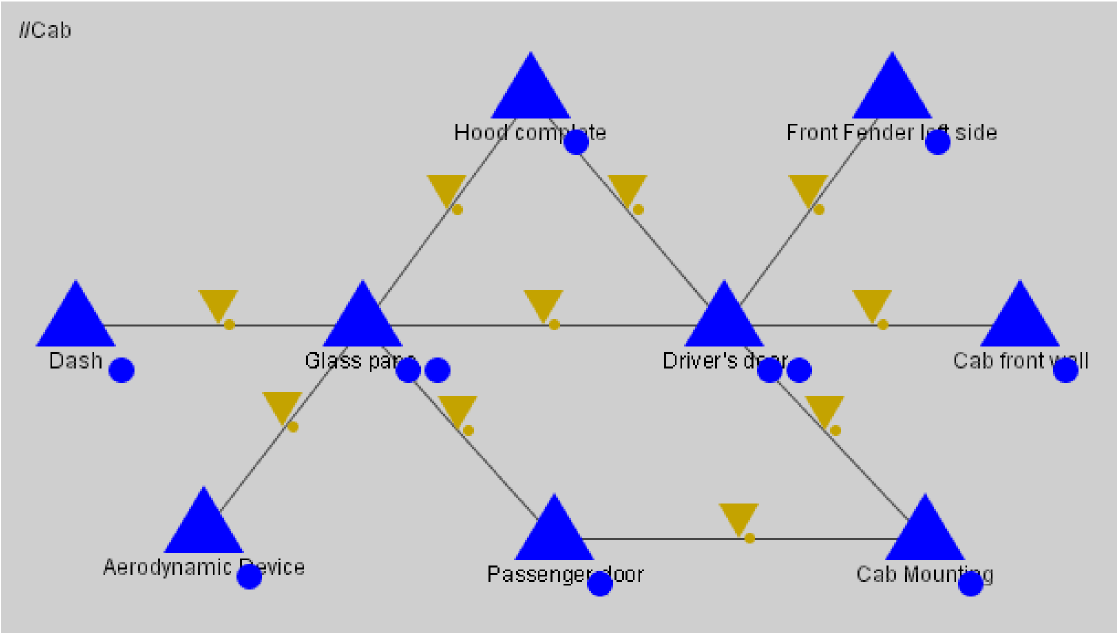


Fig. 6.20: Completed workflow

6.2.5 Decomposition of the driver's door

The driver's door is an extremely complex component rich in variance which is composed of a great number of single parts. Therefore, it is sensible to split up the part position **Driver's door**. This is realized by applying the concept of decomposition implemented in the prototype which will be described in the following. The command **decompose** is opened via the context menu of the position **Driver's door** (see Fig. 6.21).

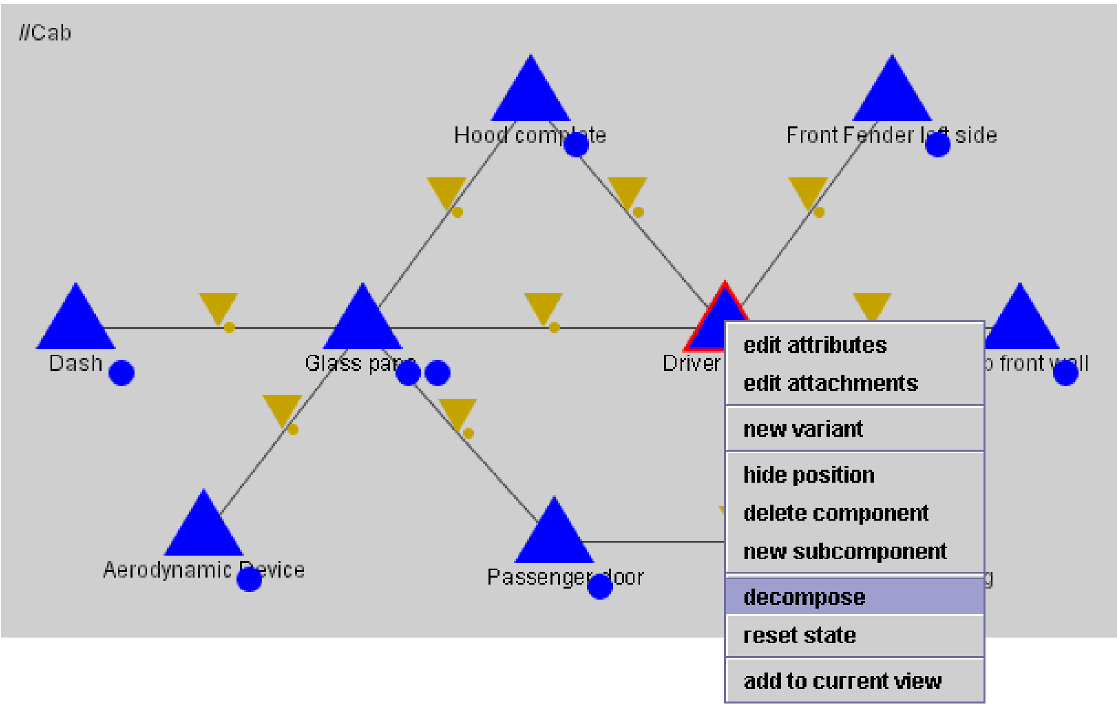


Fig. 6.21: Opening the **decompose** command

The application changes to the decomposition mode and automatically creates a position **Driver's door.1** (see Fig. 6.22).

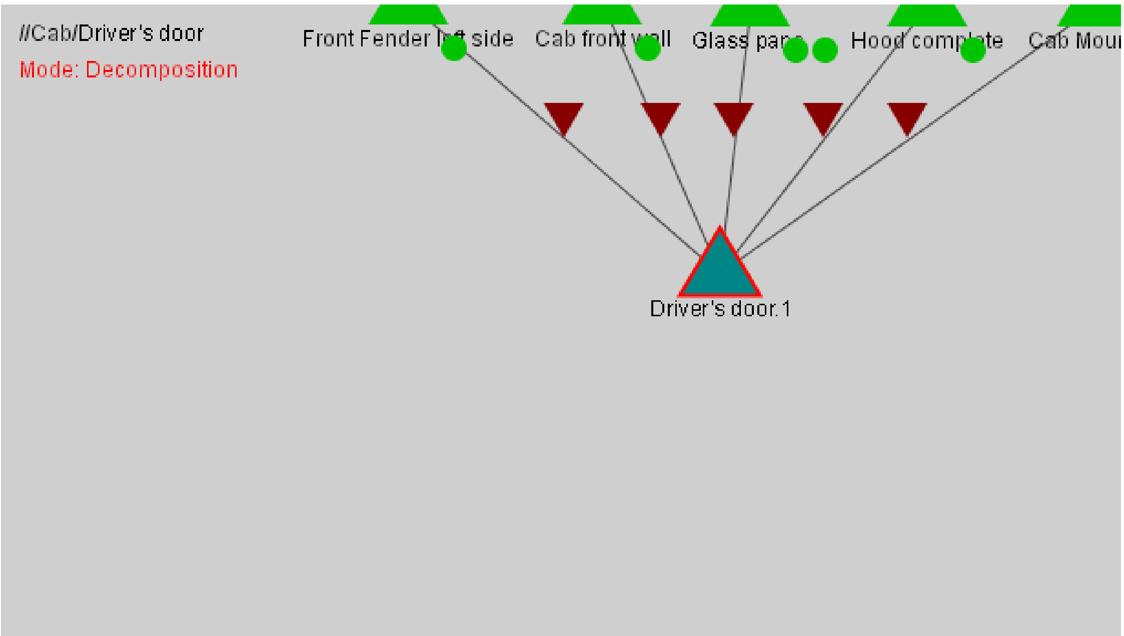


Fig. 6.22: The decomposition mode

The context menu **new component** is used to create additional positions (e.g. **planking**, **intermediate layer**, **retaining rail**, **window lift** and **guide rail**) which are connected to each other via interconnection positions. The decomposition mode is completed with the context command **carry out decomposition** (see Fig. 6.23).

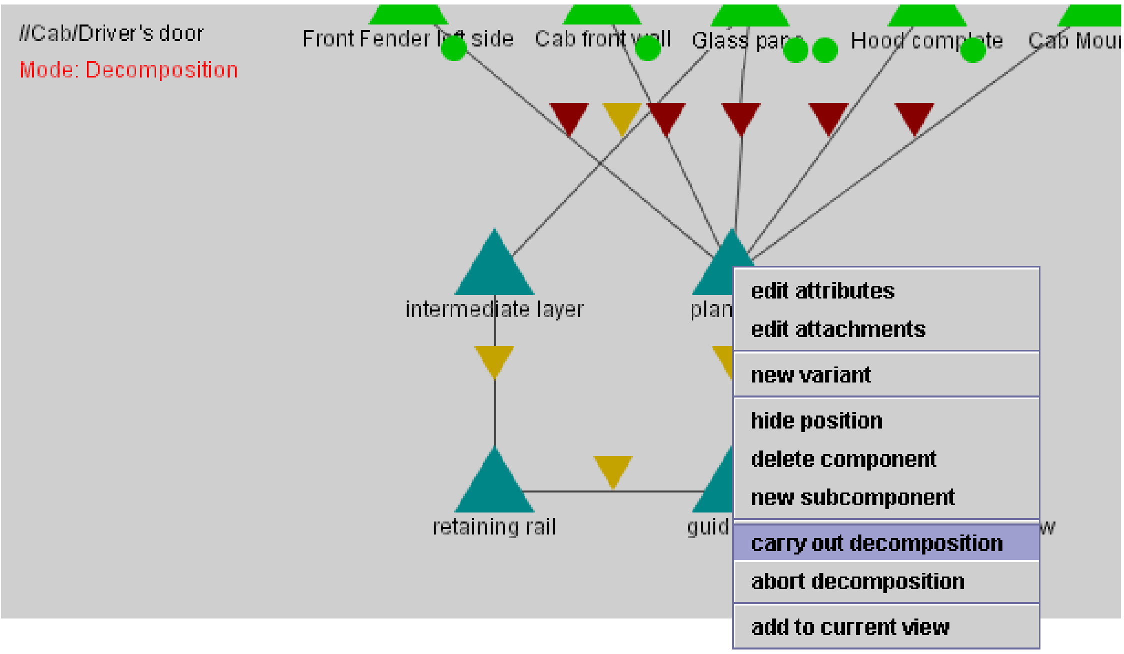


Fig. 6.23: **carry out decomposition** context command

Eventually, the results of the decomposition are summarized in the dialog box **Decompose Position**. As shown in the figure, the position **Driver's door** was split into 4 part positions and 9 interconnection positions. During the decomposition, the attributes of the position **Driver's door** are distributed to the new positions partially in an automated way. This task is carried out by the decomposing logic of the application.

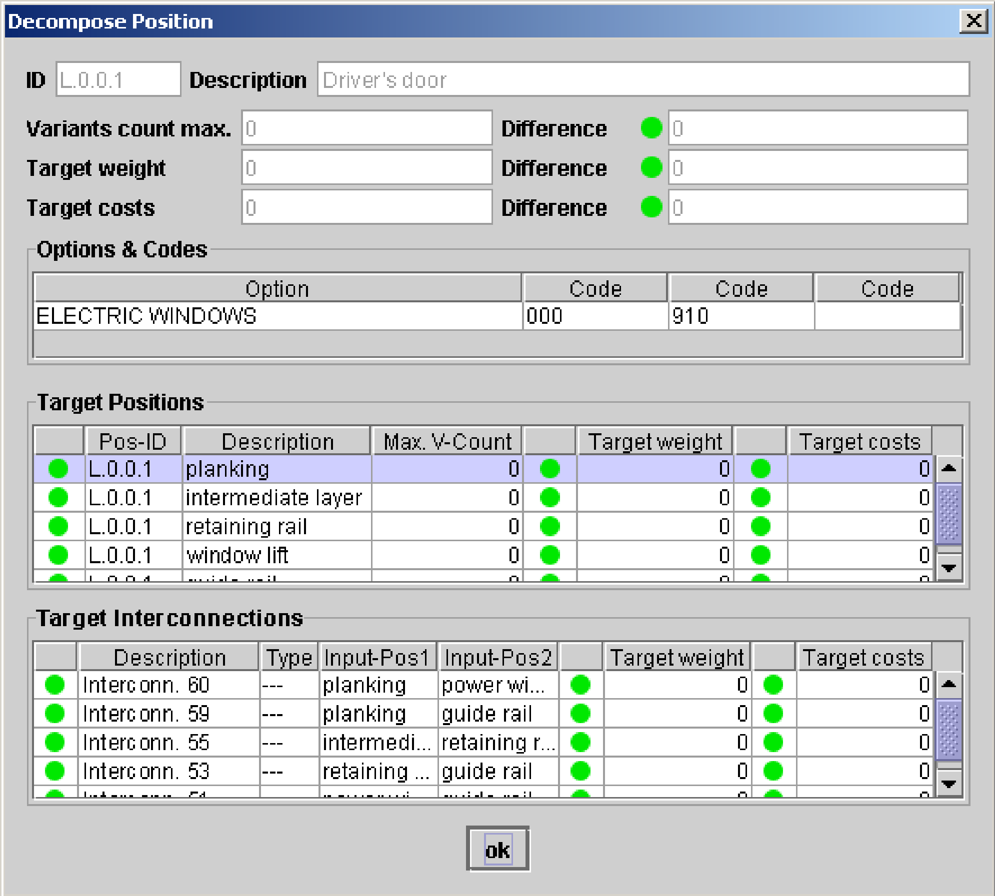


Fig. 6.24: Positions resulting from decomposition process

Some attributes of the superior position **Driver's door** were inherited to the new positions (e.g. responsibility). Other attributes, like e.g **Variants count max.**, **Target weight** and **Target costs**, were distributed evenly to the part positions.

6.2.6 Bottom-Up Wizard

The positions **window lift** and **retaining rail** shall be assembled for the "ITego" scenario in the sense of a preassembly. Due to logistics reasons, this results in the need to manage the assembly variants. In accordance with the concept of bottom-up generation introduced in section 4.5.2, Fig. 6.25 shows the dialog box of the **Bottom-Up Wizard**. For the bottom-up generation, all theoretically possible combinations of part positions are generated.

In the dialog box **Bottom-Up Result**, the assembly variants which are not feasible due to the mutually excluding code rules are displayed with a red stoplight. Similarly, the feasible assembly variants have a green stoplight (see Fig. 6.25).

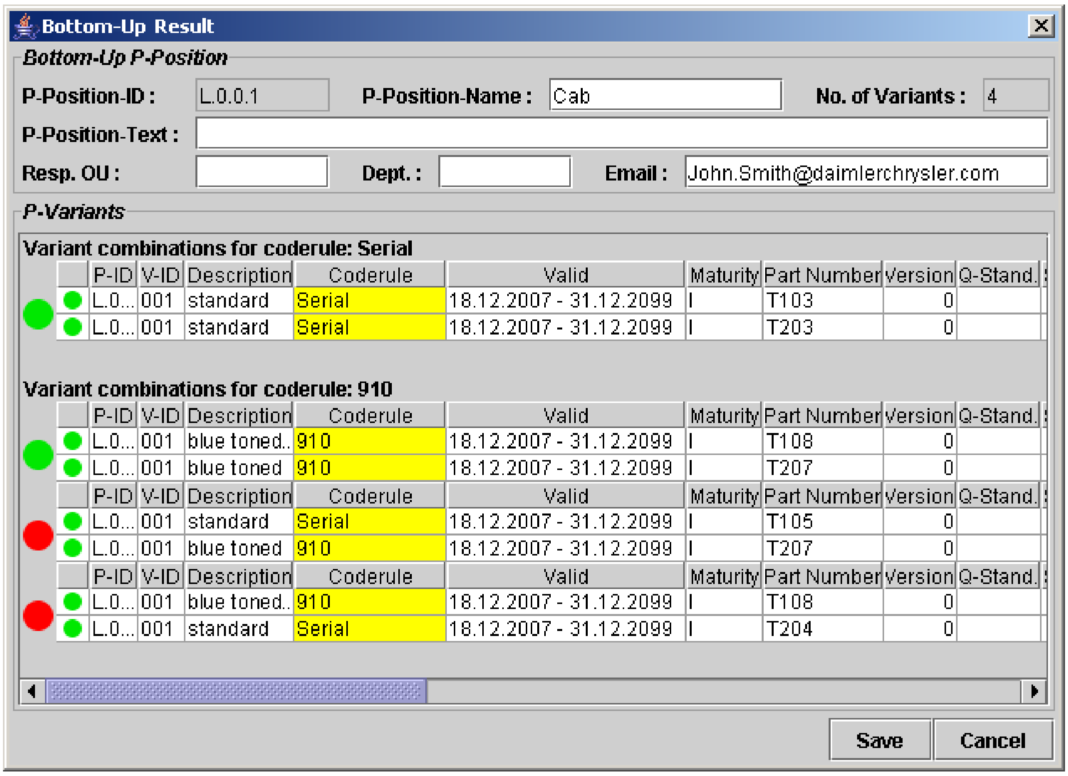


Fig. 6.25: Results of bottom-up generation

6.2.7 Appending attachments

Any number of files (of any format) can be appended to positions, position variants, interconnection positions, and interconnection position variants. The linking is realized via the **Attachments** dialog which is opened via **edit attachments** in the context menu. Fig. 6.26 shows the five attachments of the position **retaining rail**. Via **Add File** and **Remove File**, links can be created and deleted again. Files of the image formats ".jpg" and ".gif" are displayed directly in the **Picture** section. Besides, a description can be entered for each linked file which is stored in the project file.

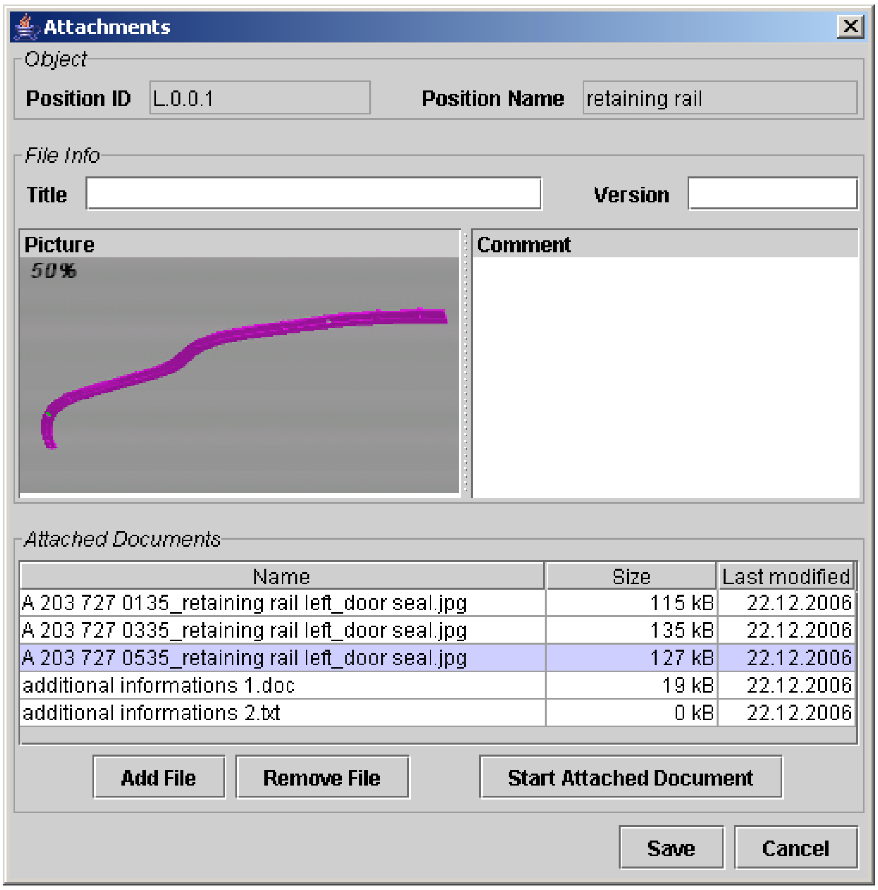


Fig. 6.26: Attachments for **retaining rail**

6.2.8 Applying the IPDM matrix

Fig. 6.27 shows the IPDM matrix of the "ITego". The right part of the matrix manages the customer orders. For the present scenario of the "ITego", three examples of customer orders **CO2**, **CO3**, and **CO1** are managed. The left top part of the IPDM matrix manages the part positions and part position variants, and the left bottom part shows the interconnection positions and their variants. In the following, the most common interactions in relation with the IPDM matrix shall be explained.

The **create customer order** button is used to create new customer orders, the **delete customer orders** button is used to cancel them. The columns **Inventory** and **Demand** document the stock and demand of the part and interconnection position variants. Based on the stock situation, customer order **CO1** is dragged before **CO3** with the left mouse button pressed.

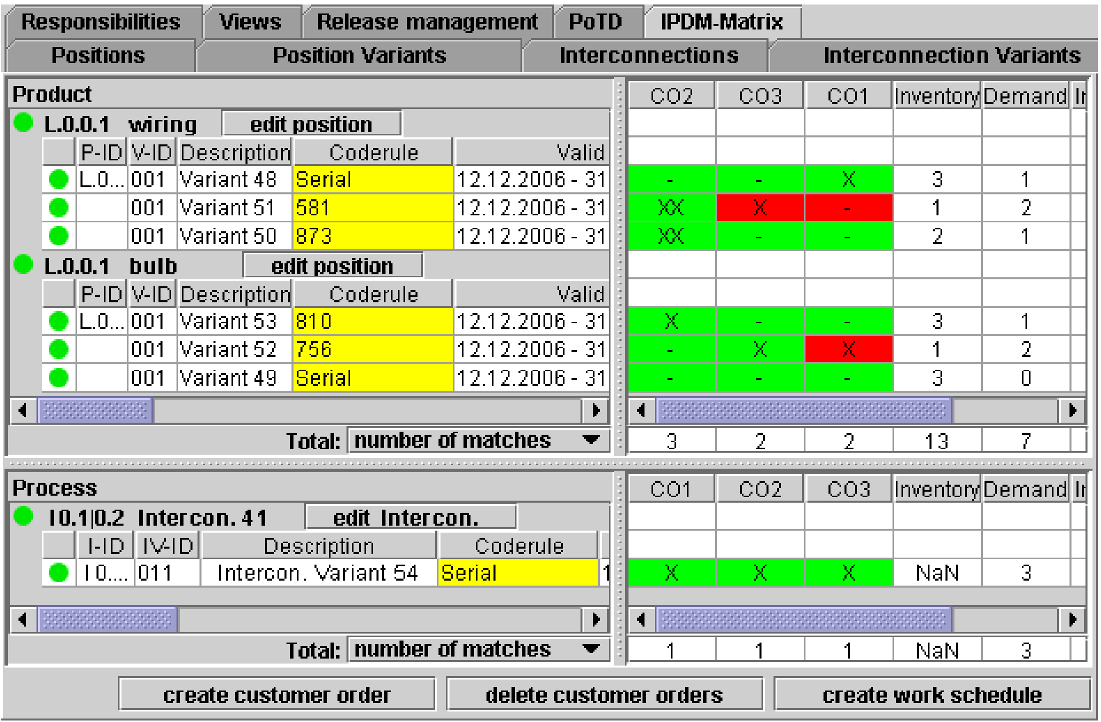


Fig. 6.27: IPDM matrix of "ITego"

Fig. 6.28 shows the result of this action. Now, customer order **CO1** can be produced without any difficulties.

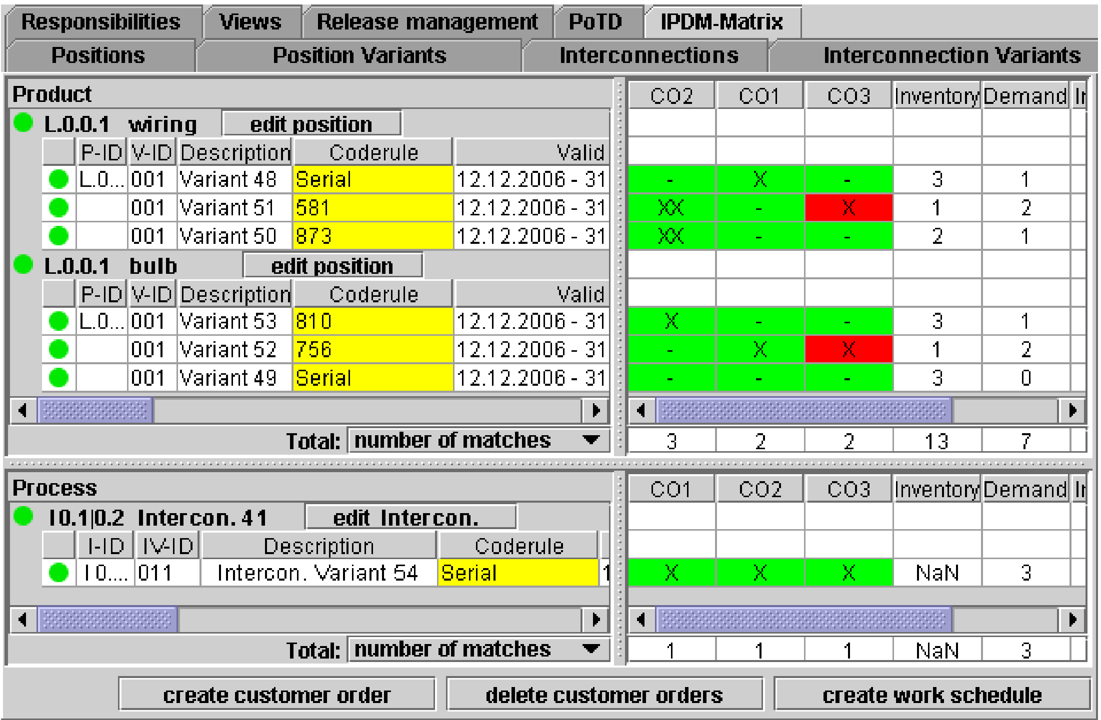


Fig. 6.28: IPDM matrix after changing the order sequence

The **create work schedule** button is used to generate customer order specific workplans (bottom-up).

7 Summary, outlook and recommendation

Starting from the current situation of industrial manufacturing companies and the resulting activities, this final chapter summarizes the results achieved in this study (cf. sections 7.1 and 7.2). Afterwards, the areas of future research which were identified during the study and the measures relevant for a productive use will be formulated in section 7.3.

7.1 Summary

The globalization of markets has had distinct effects on industrial manufacturing companies during the past years. Experience has shown that a fast reaction towards the continuously changing market and customer requirements is absolutely mandatory and critical for a sustained competitiveness. The designing of processes is gaining ever more significance. The target are flexible and continuous processes which support the entire product life cycle from product generation in engineering, procurement and production to distribution and service while being optimized with regard to cost, time, and quality. In this context, the method of product structuring and documentation is the core piece of any manufacturing company. Deeply embedded into the technical workflow organization, the product structure maps all product variants to form the basis for any process activity. The product structure has turned into a decisive competitive factor and is considered to be the most valuable asset of a manufacturing company. This is particularly true for series manufacturers with large variance as they can be found in automotive industry, for example. The degree of supporting a seamless transition from digital product engineering to the physical world of vehicle manufacturing can be interpreted as a criterion for a successful product structure. A product structure defined by product documentation is considered to be the link between the (numerous) digital views and the (single) physical representation of an individual customer vehicle. In this context, the product structuring methods used today in practice show a few weak points and do not meet the increasing requirements by far. The interconnection based product and process documentation presented in this study removes these weaknesses and considers the experience of today's methods. The study on the interconnections documentation was divided into four main parts.

The first main part of the study addressed the current situation of the manufacturing companies. The problem of variance and complex processes prevailing in the companies in relation with the affected information systems will be analyzed to reveal the extraordinary significance the product and process documentation method has in this context (see chapter 1). In the further course of this part of the study, current documentation methods were analyzed. The analysis part was concluded with the investigation of selected implementations of an integrated product and process documentation (see chapter 2).

Based on the results of this analysis phase, the second main part of the work (see chapter 3) formulated the requirements placed on the interconnection based product and process documentation. Following the procedure during the analysis, the requirements are formulated as seen from the problems of variance and process complexity.

The third main part of the dissertation (see chapter 4) described the concept of the interconnections documentation. By defining and deriving the interconnections documentation, the following concepts of the method were presented in an overview:

- definition of the overall product schema and the method to derive project specific product structure
- building and maintaining product web based on parts and interconnections
- process specific view generation via bottom-up algorithms

A detailed explanation of the substantial characteristics of these concepts was given in the various sections of this chapter. For the standardized deriving of project specific product and process structures and to support the commonality idea, the definition and structure of a product schema and a methodological procedure to derive specific product structures were described. With the definitions of the part and interconnection positions (including their variants) and the introduction of the short code rule with positive expressions, the key components of the product web (= SDE web) were introduced. With the interconnections, a new key class was introduced providing for the basis of an integrated product and process data model. In this context, a detailed methodological procedure to configure this product and process web was described. With the concept of decomposition, a methodological procedure was described which allows for a further structuring of a product and process structure derived from the product schema or define such a structure anew.

At first, a standardized procedure to develop a project specific product structure and a tool to actively manage variants and complexity based on parts and interconnections was provided.

The introduction of sequencing interconnections expanded the interconnection based product and process data model defined so far. The definitions on the preference graph and the derived manufacturing sequence provided for the prerequisite to offer an instrument to control the engineering and manufacturing process. Based on the sequencing, the method of generating assemblies was presented. Compared to conventional documentation procedures, this method offers the possibility to generate the complete variance of an assembly[1] in an automated way at any time in the product generation process.

In the further course, the mutual effects of design and manufacturing depth and their consequences on the documentation in the context of an OEM/supplier relation was displayed. The mutual effects of the technical and sales feasibility was considered in a way that specifications for a market product overview is derived based on technical restrictions.

To live up to the increasing demands of the process functions involved in the product generation process to have information in relation to tasks or functions, the concept of bottom-up algorithms to generate process specific views was presented. With the definition of additional positions (for tooling, e.g.), the interconnection based product and process data model was expanded further.

This completed the basics of the product and process data model and the description of the documentation method underlying this model. Afterwards, the integrated use of parts and interconnection positions was introduced using the change management as example. The IPDM matrix represents a logical representation of the model. The integration of the customer order and product generation process was described using a few use cases as examples. The explanation of a step-by-step migration concept to introduce the interconnection based product and process data model completed the third part of this study.

The fourth and last main part described the design and prototype implementation of the interconnections documentation (see chapter 5). At first, the object oriented design considering the Unified Modeling Language was introduced. Afterwards, the IT architectural concept used was described and the technological basics relevant for a prototype implementation were defined. Based on the realization of the IT prototype, the interconnection based product and process data model and the methodological procedure of documentation was evaluated using the business scenario "ITego" as example (see chapter 6).

The conception and prototype implementation of the interconnections documentation mean a consistent advancement of the BCT model. The new documentation method developed this way belongs to the class of the "open" parts list methods in the widest sense. With the interconnection based integration of the product and process information, the interconnections documentation provides the most efficient form of documentation.

7.2 Result of the study

Due to the enormously increased requirements placed on industrial manufacturing companies in the past years, the designing of processes has become ever more important. In this context, the designing of flexible and continuous business processes starts from the method of product structuring. While the product structure has developed into a decisive competitive factor to

1 In an expanded sense, that of the product to be manufactured as well.

become the most valuable asset of a manufacturing company, the existing methods of product and process modeling have reached their limits. Based on hierarchically oriented product structures, products are still being described as sum of single parts and major components today. Structural associativity between parts and assemblies of individual product variants or the entire product portfolio of a manufacturing company cannot be mapped with these conventional methods or to a limited extent only. Likewise, process information can also be integrated in a restricted way only.

While existing ERP and PDM system solutions start from these conventional methods, the interconnections documentation developed in this study provides the core elements of an integrated product and process data model. Products rich in variants are configured as a web of parts and interconnections. For the structuring and maintenance of the product web, a methodological procedure is available which describes the complete and efficient mapping of the product, process, and resource information of all products in a company.

With the definition of a vehicle neutral product schema which is generally valid for all product classes using the commercial vehicles division of Daimler AG as example, a standardized procedure for deriving or creating a project specific product structure was developed. This supports the reuse of parts and components.

Based on a prototype implementation, the concept of the interconnections documentation could be verified. An IT tool could be developed for the active variant and complexity management based on parts and interconnections. Using the business scenario "ITego" as an example, the procedures of how to configure a product web or react towards product specific or process modifications were explained. To use the interconnections documentation in the productive environment, the developed IT solution has to be expanded by the relevant data elements of the respective process functions. For a technical integration of the interconnections documentation or the developed IT tool into the overall IT landscape, the tool has to be expanded by a relational database. Besides, an interface to the ERP and PDM systems used today has to be realized. To allow for a deployment of the interconnections documentation which lives up to the requirements of worklife, this study describes a (step-by-step) migration strategy.

The interconnections documentation represents a consistent advancement of the method of the "complex parts lists with ruled positions" which is considered to be a quasi standard in automotive industry. While the integrated product and process documentation supports the documentation methods used in practice today, there is no other documentation method which offers cost reduction to a comparable extent.

7.3 Outlook and recommendation

When developing the interconnections documentation, special attention was paid to considering requirements and experience of corporate practice. A detailed and complete presentation of the key concept was particularly important. In this context, a migration strategy living up to practical demands was at issue while, at the same time, considering a few other aspects for the practical introduction of the interconnections documentation.

It has to be observed, for example, that the introduction of the interconnections documentation is related to the introduction of a new IT application. Experience has shown that implementing an IT system and an application method at the same time often results in difficulties. Since the methods of product and process documentation generally are very complex applications, it is recommended to use the method of the interconnections documentation for a defined scope (= pilot) at first.

While the interconnections documentation is particularly efficient for products rich in variance, there are a number of advantages for products with small variance, too. It is therefore recommended to gain a first impression with pilots with low variance. In automotive industry, this could be, e.g., a transmission model range or a less complex assembly. In accordance with the introduced migration strategy, it is advantageous for the identified data volume to restrict the managed scope to the SDE web in a first step (based on the parts and interconnection positions and their variants) including the "effectivity" function[1]. This procedure corresponds to the initial step of the described migration strategy and the updating of the left part of the IPDM matrix. Afterwards, the "pilot" documented in this way can be transferred to one of the conventional documentation methods (or in accordance with the application protocol AP214 to a standard format) via an interface. From there, the downstream processes can continue as is. A gradual expansion of the pilot up to the documentation of a complete model range can be realized any time as experience advances. With growing experience, other (downstream) process functions can be designed anew in accordance with the proposed migration strategy.

Another activity demand is required for the definition of the product schema. At present, the vast majority of manufacturing companies do not have a neutral product schema. Hence, a step-by-step procedure is recommended here, too. In a first step, the existing product structures of a manufacturing company shall be consolidated. In a second step, the consolidated product structure can then be enriched by the relevant blackbox specifications and implementation templates.

There is a concrete demand for research in the context of defining and classifying interconnections. Developing classifications for the different interface types (according to physical, logical. or electrical criteria) may distinctly reduce the documentation effort and improve documentation quality. The detailing of the interconnections on the feature level in particular and the definition of corresponding interconnection libraries seem to be promising areas of research.

1 For parts demand calculation, effectivity is absolutely mandatory and part of the key term.

The largest potential, however, is in the design of the different process functions in the different areas of application. There is no other documentation method which offers so much cost reducing potential.

Glossary

IT terms (amongst others UML, OMG, modeling)

Computer Aided Software Engineering tools: According to [WiCase], these tools are programs which support software engineers in planning, designing and documenting software. An important component of CASE tools is a graphical way of notation which is used to visualize architecture of the software system.

Extensible Markup Language: The *eXtensible Markup Language XML* is a standard for the generation of machine and human readable documents in the form of a tree structure. XML defines the rules for the structure of such documents. XML can be used to define other languages making it a meta language.

Graphical User Interface: A *Graphical User Interface GUI* is an interface between a computer and a person which allows an interaction between them via graphical components using input devices (like e.g. mouse, keyboard, etc.).

Meta model: A model can also be considered to be a system. If such a system is modeled, a meta model of the model is obtained. Hence, the meta model describes the related model and is located on a higher level of abstraction than the model. The meta model, in turn, can be described again with a meta model (i.e. a meta meta model), a process which can be repeated arbitrarily.

Model: A model is an abstract view of a system and always has to be viewed in the context of this system. Typically, the system is viewed with a specific focus, i.e. a model only describes portions of a system and neglects characteristics which seem to be unimportant under a certain aspect. Frequently, models are domain specific (see Domain), but do not have to be domain specific (see UML). The typical key characterisitcs of a model are a high degree of abstraction and platform independence.

Object Management Group: The *Object Management Group* is a standardization pool in the area of object oriented programming. It mainly deals with the development of standards to facilitate a model based software development and make it compatible. Known standards and "visions" of the OMG are: MDA, UML, UML Profiles, OCL, MOF, QVT, XMI.

Platform: In the context of computers, a platform means a system on which a computer program (software) is executed. Typically, it refers to a combination of operating system and hardware where the software is executed.

Semantics: Semantics means the meaning or statement of a model element type. Since the model element types are determined in the meta model already, the semantics mostly is defined at this (early) time, too. The semantics can be stated in a natural language or by OCL. It is already predefined for the elements from the UML models.

Syntax: The syntax describes the gammar of a model, i.e. it defines the models which are accepted by a model parser. [Birchm02] differentiates between the concrete and the abstract syntax: "The concrete syntax is in a form that can be read and understood by humans, while the abstract syntax primarily addresses computers. A decisive characteristic is the degree of formalism. To make model transformation capable of being automated, the syntax at least must be understandable to the computer, i.e. of a formal nature."

Unified Modeling Language: *Unified Modeling Language* is a modeling language (or collection of modeling languages) which is mainly used for the modeling of object oriented software systems. It provides graphical notations in the form of diagrams which makes it a Visual Modeling Language. To be precise, UML is rather a meta modeling language since it can be used to create and describe own models. Nevertheless, the term of a modeling language is not wrong, since each meta model is a model at the same time. From version 1.1, the UML has been standardized by the OMG which has advanced the language (up to current version 2.1).

Visual Modeling Language: A *Visual Modeling Language* uses a graphical notation syntax. The UML diagrams are likely to be the most wellknown examples of a VML. But there are also text based modeling languages, like e.g. MOF, OCL, etc. Annotation: Programming languages are not considered to be modeling languages, since they typically depend on a specific platform and contain technical aspects. Besides, they cannot be assigned to a clearcut subject and their degree of abstraction is rather low (cf. Model).

XML Metadata Interchange: *XML Metadata Interchange* is an OMG standard. To a growing extent, it is used in the context of model based software engineering as exchange format between software development tools. The data exchange of objects is based on meta meta models according to the Meta Object Facility (MOF). Apart from UML models, any meta data can be exchanged as long as they can be expressed using the MOF.

Terms of product structuring

To be able to communicate the unimaginable number of products of a car manufacturer to customers, a systematic reviewing of the products is carried out regarding a clear hierarchical structuring. In the following, the known terms of model range, model subrange, type, model and variant are described.

Model: Model means the connection of vehicle type and steering (cf. [Fische93]). While the type "C200 Estate" offered in Germany means a car with left-handed steering, the identical type in Great Britain means right-handed steering. Like the type, the model is not a product in its true sense (cf. [Ohl00]).

Model range: A model range does not represent a product in its true sense, but summarizes similar products in a collective term (cf. [Herlyn90]). In automotive industry, the products belonging to a model range consist of identical parts to a high (cf. [Merced95c]). Hence, model ranges are frequently used to structure sales documents or parts lists. This seems especially sensible in situations where the actual number of offered products is very large. The term of a model range is not only used in relation to vehicles as vehicle model range, but also in connection with engines and transmissions as engine or transmission model range. The engines of a specific engine model range, for example, can be used in vehicles of different vehicle model ranges.

Model subrange: In addition to model ranges, the automotive industry also uses the term of a model subrange. To means the connection of a vehicle model range to a specific bodywork (cf. [Heinis91]).
Model subranges also exist for engines. As for the bodywork in vehicle model ranges, displacement, for example, represents a model subrange for engines. It is the combination of an engine model range with a specific engine variant that makes up a single engine.

Type: In automotive industry, the cross product of vehicle and engine model subranges is called type. If a specific bodywork of a vehicle model range is combined with a variant of an engine model range, a type is created. Not all possible combinations of bodyworks of a vehicle model range with a variant of an engine model range are actually offered as vehicle type. In general, car manufacturers use the type as sales term which links to a concrete selling price for the first time (cf. [Ohl00]).

Variant: (Product) variants are characterized by fulfilling a unified main function with different characteristics and different additional functions. As example of a unified main function in the context of automotive industry is a specific vehicle model range. The different characteristics are expressed in attributes like form of bodywork or engine variant. The different additional functions are expressed in the optional equipment characteristics of the respective vehicle which define the individual product, i.e. the actual product variant, in the form of series and optional equipment. Hence, variants are end products which differ in subordinate characteristics, but not (or little) in their basic structure. In this study, the term (product) variant always refers to saleable products.

Manufacturing levels in automotive industry

In the following, the different manufacturing levels and their distinction are described. For further definitions, the reader shall be referred to literature (cf. et.al. [Lingna94], [DIN_77], [VDIT77_76], sim. [Ungehe86], [Heinis91], definitions in [VDIT77_76]).

Assembly: Assemblies are objects consisting of two or several (single) parts and/or assemblies of a lower level (cf. [Ohl00]). From the use-related view, an assembly is either a direct constituent of a product, major component, or assembly of a higher level.

Major component: This term is used very frequently in automotive industry for complex assemblies. While products are offered in the sales market for use, major components are "internal products" which are not offered to customers for purchasing (cf. [Lingna94]). Thus, major components are in a intermediate position between the product on one side and the assembly on the other. In automotive industry, complex assemblies, like e.g. engines, transmissions and axles are called major components (cf. [Herlyn90], [Ohl00]).

Part (single): According to [DIN_76], a single part is a part which cannot be decomposed destruction free (cf. [VDIT77_76], , sim. [Ungehe86]). From the user's point of view, there is no need for a further decomposition (cf. [Heinis91]). From the use-related view, a part is a direct constituent of a product, major component, or assembly. This study refrains from a detailed differentiation of part, single part and finished part. The terms are rather used as synonyms.

Product: A product means a working object created in production whose purpose is the use in the sales market (cf. [Lingna94], [Ohl00]). Products are objects comprised in themselves which consist of a defined number of assemblies and/or parts [Ungehe86]. In the automotive industry context, a product is a concrete vehicle produced for the sales market which is offered to a customer for purchasing.

Rough part: Rough parts are objects which have been processed, but do not yet meet the specifications of drawing and design (cf. [Herlyn90]). In an abstract way, rough parts parts can be called parts both from a physical and use-related view (cf. [Lingna94]).

Semi-finished part: The term semi-finished part means all items which enter a manufacturing unit in an unmachined condition. In an abstract way, semi-finished parts parts can be called parts both from a physical and use-related view (cf. [Lingna94]).

List of abbreviations

<u>Abbreviations used in the text</u>

ACT	Activity
CAD	*Computer Aided Design*
CAM	*Computer Aided Manufacturing*
CAP	*Computer Aided Planning*
CAPP	*Computer Aided Process Planning*
CAQ	*Computer Aided Quality*
CMP	Component
COP	Customer order process
cPDM	*collaborative Product Data Management*
DD	Design depth
EDM	*Engineering Data Management*
ePLM	*electronic Product Lifecycle Management*
ERP	*Enterprise Resource Planning*
FLO	Factory layout
GUI	*Graphical User Interface*
IDE	*Integrated Development Environment*
IPDM	Integrated Product and Process Data Model
iPPE	*integrated Product and Process Engineering*
iPS	integrated Planning System
MBC	Mercedes-Benz Cars
MD	Manufacturing depth
OEM	*Original Equipment Manufacturer*
OMG	*Object Management Group*
OPS	*Overall Product Schema*
PDC	*Product Definition and Commerce*

PDM	*Product Data Management*
PGP	Product generation process
PPM	integrated product and production data model
PPS	Produktionsplanung und -steuerung (production planning and controlling)
PSM	*Platform Specific Model*
psPS	project specific Product Schema
PTA	Package of technical associativity
PVS	Product variant structure
SDE	Smallest design element
SDK	*Software Development Kit*
UML	*Unified Modeling Language*
VMEA	*Variant Mode and Effects Analysis*
VPDM	*Virtual Product Definition Management*
XML	*EXtensible Markup Language*

Abbreviations used in the figures

A	(A)ssembly
C	(C)ode
CR	(C)ode (r)ule
IP	(I)nterconnection (p)osition
IPV	(I)nterconnection (p)osition (v)ariant
M	Model
MC	(M)ajor (c)omponent
MP	(M)odular (p)arts list
Op	Operation
P	(P)roduct
P_POS	Parts list position
POS	Position
PP	(P)art (p)osition
PPV	(P)art (p)osition (v)ariant
T	Par(t)
WP	(W)ork(p)lan

List of figures

List of tables

Bibliography

[Adam98] Adam, D.; Johannville, U.; *Die Komplexitätsfalle;* In: Komplexitätsmanagement, Ed.: Adam, D., Gabler, Wiesbaden; 1998.

[Alam02] Alam, D.; Kühn, T.; *Produktdokumentationshandbuch;* Internal workpaper, DaimlerChrysler AG, Böblingen; 2002.

[Anderl94] Anderl, R.; Wasmer, A.; *Chancen und Risiken in der Produktdatentechnologie;* In: Gausemeier, J. (ed.): Produktdatenmodellierung und Prozessmodellierung als Grundlage neuer CAD-Systeme, pp. 58-92, Hanser, Munich/Vienna; 1994.

[Audi96] Audi (ed.); *Price List: A4-A6-S6-A8-S8-Cabriolet;* Audi AG, Ingolstadt; 1996.

[AutKol90] Autorenkollektiv; *Strategien auf dem Weg zur Fabrik von morgen;* In: Wettbewerbsfaktor Produktionstechnik; Aachen Colloquium on Machine Tools, pp. 3-34, Düsseldorf; 1990.

[Balzer05] Balzert, H.; *UML 2 in 5 Tagen - Der schnelle Einstieg in die Objektorientierung;* W3L-Verlag, Bochum; 2005.

[Bastia94a] Bastian, J.; *Unterlagen zur Auftragsanalyse, Codeprognose und zu Planaufträgen;* Workpaper, Mercedes-Benz AG, Stuttgart; 1994.

[Bastia94b] Bastian, J.; *Grobkonzeption: Neues Prognose-Modell (NPM);* Workpaper, Mercedes-Benz AG, Stuttgart; 1994.

[Bedi08] Bedi, P.; *Überlegungen zum Vergleich der Anzahl der Schlüsselbegriffe;* Internal workpaper, Daimler AG, Fellbach; 2008.

[Bender04] Bender, K.; *Produktdaten Management (PDM) - Vorlesungsunterlagen Teil 1;* Chair of Information Technology in Engineering, Technical University Munich; 2004.

[Berg93] Berg, H.; *Motorcars – Between Growth and Protectionism;* In: H.W. de Jong (ed.): The Structure of European Industry, 3rd edition, pp. 121-146, Dordrecht/Boston/London; 1993.

[Birchm02] Birchmeier, Peter; *Model Transformation;* Seminar in Software Engineering, Informatics Institute, Zurich University, May 2000. http://seal.ifi.unizh.ch/~pinzger/Birchmeier_Peter_Model_Transformation.pdf

[Bley97] Bley, H.; Wuttke, C. C.; *Logistikgerechte Produktentwicklung durch Kopplung von Produkt- und Produktionsmodellierung;* In: VDI Reports Vol. 1357 "Neue Generation von CAD/CAM-Systemen – erfüllte und enttäuschte Erwartungen", pp. 243-262, VDI, Düsseldorf; 1997.

[Bliese92] Bliesener, M.-M.; *Bestandssenkungspotential durch Einsatz moderner Verfahren der dynamischen Materialdisposition im Fertigungsbereich, dargestellt an einem praktischen Beispiel aus der Automobilindustrie;* In: Betriebswirtschaftliche Forschung und Praxis, w/o year, Vol. 5, pp. 456-472; 1992.

[BMW96] BMW (ed.); *Prices BMW Passenger car range; As of: March 1, 1996;* BMW AG, Munich; 1996.

[Bochtl96] Bochtler, W.; *Modellbasierte Methodik für eine integrierte Konstruktion und Arbeitsplanung;* Dissertation at the Faculty for Machinery of the Technical College Aachen, Shaker, Aachen; 1996.

[Brändl94] Brändli, N.; *Neue Ereignisdokumentation im Geschäftsbereich Pkw der Mercedes-Benz AG (NED-GBP) – Fachkonzept Daten;* Internal workpaper, Daimler-Benz AG, Ulm; 1994.

[Brändl95] Brändli, N.; Dumschat R.; *Regelbasierte Stücklisten im Anwendungsdatenmodell ISO10303-214;* Produktdaten Journal, 2, Vol. 1, pp. 10-12; 1995.

[Briest93] Briest, J.; Heuser, M.; *Einsatz eines Variantenkostenmodells;* Lecture papers, Technology Colloquium of Volkswagen plant Salzgitter, Volkswagen AG, Salzgitter; 1993.

[Bruhn01] Bruhn, M.; *Produktmanagement;* Lecture papers, Basel University; 2001.

[Buchte02] Buchter, A.; Klose, H.; *SMARAGD Benutzungshandbuch (Version 4.3.1) – Konfigurations-Management & -Versionierung;* SMARAGD project management at DaimlerChrysler AG, Stuttgart; 2002.

[Bullin96] Bullinger, H.-J.; Warnecke, H.-J. (ed.); *Neue Organisationsformen im Unternehmen;* Springer, Berlin; 1996.

[Burkha99] Burkhardt, R.; *UML - Unified Modelling Language. Objektoriehtierte Modellierung in der Praxis (2. Auflage);* Addison-Wesley; 1999.

[Christ97] Christmann-Jacoby; Bongardt; Maas; *Kommentierung der Methode IPDM;* Internal workpaper, DaimlerChrysler AG, Bremen;1997.

[Daimle00] DaimlerChrysler AG; *IPDM Strategiepapier;* Internal workpaper, DaimlerChrysler AG; 2000.

[DIN_77] Deutsches Institut für Normierung e.V. (DIN) (ed.); *DIN 199, Part 2: Terminology associated with technical drawings and parts lists; Parts lists;* Beuth, Berlin/Cologne; 1977.

[Eigner01] Eigner, Martin; Stelzer, Ralph; *Produktdatenmanagement-Systeme - Ein Leitfaden für Product Development und Life Cycle Management;* Springer, Berlin/Heidelberg; 2001.

[Eisenh02] Eisenhart v. Rothe, M.; *Konzeption und Einführung eines IT-gestützten Produkt-Konfigurationsmanagements für die technische Information in der Automobilindustrie und -herstellung;* Dissertation at the Technical University Clausthal 2002, Shaker, Aachen; 2002.

[Eversh93] Eversheim, W.; Kümper, R.; *Variantenvielfalt in der Produktion – Ursachen, Lösungsansätze und Erfahrungen;* Technical Report, Laboratory for Machine Tools and Business Operations (WZL) of RWTH Aachen University, Aachen; 1993.

[Ewers94] Ewers, K.-C.; *Regelbasierte Organisation hochvarianter Produktstrukturen auf der Basis moderner Datenbank-Technologie;* Diploma thesis at the Faculty for Informatics of Ulm University in cooperation with Daimler-Benz AG, Research & Technology, Ulm; 1994.

[Faux94] Faux, I.; *Integrierte Produkt- und Prozessmodellierung;* In: VDI Reports Vol. 1148, pp. 17-37, VDI, Düsseldorf; 1994.

[Fische93] Fischer, U.; *Konzipieren einer Methode zur Simulation der Abhängigkeiten, Auswirkungen und der Realisierbarkeit von Programmänderungen in einem Automobilunternehmen mit*

Entwicklung eines Simulationsprototyps zur Unterstützung der ergebnisorientierten Produktions- und Vertriebsplanung; Diploma thesis at Esslingen Technical College, Department of Mechanical Engineering/Manufacturing Systems, Esslingen/Stuttgart; 1993.

[Frohma88] Frohnmayer, B.; *Die Technische Dokumentation der Erzeugnisse in der Daimler-Benz AG;* Internal report; 1988.

[Garwoo97] Garwood, D.; *Bills of material: structured for excellence;* 5th ed., Marietta Dogwood Publishing Company; 1997.

[Gausem00] Gausemeier, J.; Anderl, R.; Kespohl, H.; Claassen, E.; Avenarius, J.; *Integrationsplattform für die Entwicklung mechatronischer Systeme;* Proceedings CAD 2000; 2000.

[Grabow94a] Grabowski, H.; Erb, J.; Polly, A.; Anderl, R.; *STEP – Grundlage der Produktdatentechnologie – Aufbau und Entwicklungsmethodik;* In: CIM Management 4/94, pp. 45-51, Oldenbourg; 1994.

[Grabow94b] Grabowski, H.; Erb, J.; Polly, A.; Anderl, R.; *STEP – Grundlage der Produktdatentechnologie (Teil 2) – Das integrierte Produktmodell;* In: CIM Management 5/94, pp. 36-43, Oldenbourg; 1994.

[Grabow95] Grabowski, H.; Meis, E.; Hain, K.; *Integriertes Produkt- und Produktionsmodell;* In: Informationstechnik und technische Informatik 37, Vol. 5, pp. 32-35; 1995.

[Grabow99] Grabowski, H.; *Rechnerunterstütztes Konstruieren und Erstellen von Fertigungsunterlagen (CAD) II;* Lectures of Interfacultative Institute for Informatics Applications at the Faculty for Informatics, Karlsruhe University; 1999.

[Groll99] Groll, M.; *IPDM Processes and Data;* Internal workpaper, DaimlerChrysler AG; 1999.

[Grupp76] Grupp, B.; *Elektronische Stücklistenorganisation in der Praxis;* Forkel, Wiesbaden; 1976.

[Grupp89] Grupp, B.; *Stücklisten- und Arbeitsplanorganisation mit Bildschirmeinsatz;* 4th ed., Forkel, Wiesbaden; 1989.

[Grupp95] Grupp, B.; *Aufbau einer optimalen Stücklistenorganisation - offene Stücklisten, Variantengenerator, PPS-Rahmen, CAD-Connection, Praxisbeispiele;* expert verlag, Renningen-Malmsheim; 1995.

[Haasis95] Haasis, S.; *Wissens- und featurebasierte Unterstützung der Konstruktion von Stirnradgetrieben unter besonderer Berücksichtigung des Gussgehäuses;* Dissertation at the Esslingen Technical College, VDI, Düsseldorf; 1995.

[HamiNo07] Hami Nobari, S.; *Methode zur wirkungsorientierten Beschreibung variantenreicher Erzeugnisse (WIBER) - Handhabung der Komplexität in der technischen Erzeugnisdokumentation;* Dissertation at Faculty V, Traffic and Machinery Systems, of Technical University Berlin, Shaker, Aachen; 2007.

[Heinis91] Heinisch, M.-E.; *Entwicklung eines Konzeptes für die Erstellung und Realisierung einer ergebnisorientierten Programmplanung am Beispiel der Automobilindustrie;* Diploma thesis at the Faculty of Economics and Social Sciences of Friedrich-Alexander-University Erlangen-Nuremberg in cooperation with Mercedes-Benz AG, Nuremberg/Stuttgart; 1991.

[Heinri96] Heinrich, J.; *Prognose von Sonderausstattungen – Klassifizierung von Zeitreihen;* Diploma thesis at the Faculty for Mathematics and Economics Sciences of Ulm University in cooperation with Daimler-Benz AG, Research and Technology, Ulm; 1996.

[Herlyn90] Herlyn, W.J.; *Zur Abbildung variantenreicher Erzeugnisse in der Automobilindustrie;* VDI

Progress Reports, Vol. 16, No. 52, VDI, Düsseldorf; 1990.

[Hirzel95] Hirzel, J.; *Programmplanung für variantenreiche Produkte im Wandel vom Verkäufer- zum Käufermarkt;* In: G. Bol, T. Christ, B. Suchanek (ed.): Das Potential der Lean-Techniken in der Kraftfahrzeugwirtschaft - Aktuelle Entwicklungen der Material-Logistik in Theorie und Praxis. Materials for Automotive Industry, VDA Series, Vol. 6, pp. 49-61; 1995.

[Hübner75] Hübner, T.; *Produktionsprogrammplanung bei dezentraler Fertigung unter besonderer Berücksichtigung der Verhältnisse in der Automobilindustrie;* Dissertation, Braunschweig Technical University, Braunschweig; 1975.

[Iloge01] Iloge et al.; *Handbuch der Prozessgestaltung;* Internal workpaper, DaimlerChrysler AG; 2001.

[Jeckle03] Jeckle, Mario; Rupp, Chris; Hahn, Jürgen; Zengler, Barbara; *UML 2 Tutorial - Einführung in die neue Standardmodellierungssprache;* Net. ObjectDays 2003, Erfurt, p. 93, September 2003. http://www.jeckle.de/files/nod2003tutorial.pdf

[Johnso96] Johnson, R.; Leonard, M.; *PDM Benefits Analysis Guide – Assessing Product Data Management Systems for Engineering and Manufacturing;* Springer, Berlin; 1996.

[Jüngst04] Jüngst, W.; *Concept "Ports & Connections" IPDC;* Internal workpaper, DaimlerChrysler AG; 2004.

[Katzen02] Katzenbach, A.; *Gedanken zu EDM – Überlegungen zur Weiterentwicklung der EDM-Strategie;* Internal workpaper, DaimlerChrysler AG, Ulm; 2002.

[Kurbel95] Kurbel, K.; *Produktionsplanung- und -steuerung: Methodische Grundlagen von PPS-Systemen und Erweiterungen;* 2nd improved edition, Oldenbourg, Munich/Vienna; 1995.

[Landvo94] Landvogt, J.; *Entwicklung eines verbesserten Prognosemodells für die kundenspezifische PKW-Produktion;* Diploma thesis at the Institute for Materials Handling/Logistics of Technical University Berlin in cooperation with Mercedes-Benz AG, Berlin/Stuttgart; 1994.

[Lingna94] Lingnau, V.; *Variantenmanagement – Produktionsplanung im Rahmen einer Produktdifferenzierungsstrategie;* Erich Schmidt, Berlin; 1994.

[Lorenz00] Lorenz, D.; *Integriertes System im Maschinenbau – Integration von EDM/PDM, PPS/ERP und Automation im Maschinenbau;* In: Industrie Management, Vol. 16/2000, p. 81, GITO, Munich; 2000.

[Maeker97] Maeker, M.; *Seite an Seite mit dem Produkt – Workflow-Systeme übernehmen das Management aller produktbeschreibenden Daten und steuern die Prozesse;* In: Arbeitsvorbereitung, Vol. 1, pp. 66-68; 1997.

[Mann93] Mann H.; Kröger U.; *Neue Erzeugnisdokumentation in der Mercedes-Benz AG: Darstellung der Methode "Dokumentation mit geregelter Position" in einem Prototyp;* Workpaper, Mercedes-Benz AG, Stuttgart; 1993.

[Martin88] Martin J.; *GVS gegen die Varianten-Planungsgrenze;* In: Online, w/o year, Vol. 5, pp. 398-404 and Vol. 6, pp. 496-499; 1988.

[Matt04] Matt, E.; *Produkt-Dokumentation mit DIALOG – Leistungskatalog (Level 1-3);* Internal workpaper, DaimlerChrysler AG; 2004.

[Maximi07] Maximilian, G.; *Lego-Truck: Teile und Verbindungen;* Internal workpaper, Filmstudio GBG; 2007.

[Meinin94] Meininger, K.-U.; *Abstraktionsbasierte Bereitstellung bereichsübergreifender Planungsdaten für die Produktionsplanung bei Serienfertigung variantenreicher Erzeugnisse;* Schulz-Kirchner, Idstein; 1994.

[Merced94] Mercedes-Benz (ed.); *Handbuch der Produktdokumentation;* Internal training document, Mercedes-Benz AG, Stuttgart; 1994.

[Merced95a] Mercedes-Benz (ed.); *Vielfaltmanagement;* Mercedes-Benz AG, Stuttgart; 1995.

[Merced95b] Mercedes-Benz (ed.); *DIALOG – Ergänzungswerk: Die neue Produktdokumentation;* Mercedes-Benz AG, Stuttgart; 1995.

[Merced95c] Mercedes-Benz (Hrsg.); *DIALOG – Für mehr Verständnis: Hintergründe, Sinn und Zweck von DIALOG, der neuen Produktdokumentation im Geschäftsfeld PKW;* Mercedes-Benz AG, Stuttgart; 1995.

[Merced96a] Mercedes-Benz (Hrsg.); *Price list – Passenger cars and special equipment. Effective from June 3, 1996;* Mercedes-Benz AG, Stuttgart; 1996.

[Merced96b] Mercedes-Benz (Hrsg.); *Der Weg zur optimalen Teilevielfalt;* Mercedes-Benz AG, Stuttgart; 1996.

[Monke01] Monke, E.; *IPDM – Komplexitätsbeherrschung durch Methode;* Internal workpaper, DaimlerChrysler AG; 2001.

[Monke95] Monke, E.; *Gedanken zur Prognose auf Basis von Coderegeln;* Workpaper, Mercedes-Benz AG, Bremen; 1995.

[Ohl00] Ohl, Stefan; *Prognose und Planung variantenreicher Produkte am Beispiel der Automobilindustrie;* VDI Progress Reports, Vol. 16, No. 120, VDI, Düsseldorf; 2000.

[Ott97] Ott, T.; Kretzschmar, O.; Goldstein, B.; Kiesling, G.; *Integrierte Modellierung von Prozeß und Produkt im Entwicklungsbereich;* In: IM Informations Management & Consulting, Vol. 12 Special edition 1997, pp. 8-14; 1997.

[Ottens96] Ottens, T.; *Machbarkeitsstudie über die Prognose ausgewählter Kombinationen von Sonderausstattungscodes;* Diploma thesis at the Faculty of Economics and Social Sciences of Friedrich-Alexander-University Erlangen-Nuremberg in cooperation with Mercedes-Benz AG, Nuremberg/Stuttgart; 1996.

[Pahl05] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.; *Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung;* 6th ed., Springer, Berlin/Heidelberg/New York; 2005.

[Paul86] Paul, J.; *Planung, Steuerung und strategische Ausrichtung der Produktentwicklung;* RKW-Handbuch Forschung und Entwicklung, Konstruktion, Bd. 3, E. Schmidt-Verlag, Berlin; 1986.

[Pfeiff89] Pfeiffer, W.; Dörrie, U.; Gagstetter, S.; Wiegland, C.; Gerharz, A.; *Kosten der Typenvielfalt;* Research and Work Report No. 13 of Research Group for Innovation and Technological Forecasts (FIV) at Department for Industrial Business Management of the Faculty of Economics and Social Sciences of Friedrich-Alexander-University Erlangen-Nuremberg in cooperation with Mercedes-Benz AG, Nuremberg; 1989.

[Piller03] Piller, T.; Stotko, C.; *Mass Customization und Kundenintegration – Neue Wege zum innovativen Produkt;* Symposion, Düsseldorf; 2003.

[Ploenz98] Ploenzke CSC (ed.); *Produkte und Prozesse gestalten für die Märkte von morgen;* 7th

International Congress Life Cycle Management, Mainz; 1998.

[Rathno93] Rathnow, P.J.; *Integriertes Variantenmanagement - Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt;* Vandenhoeck & Ruprecht, Göttingen; 1993.

[Roh98] Roh, H.; *Produktionsplanung und -steuerung;* PPS 3601, CDR5; September 1998.

[Römer06] Römer, S.; *Entwicklung eines modellgetriebenen Ansatzes zur Generierung von Struktur und Ablaufsteuerung grafischer Benutzeroberflächen;* Diploma thesis at the Faculty for Informatics of Ulm University in cooperation with eXXcellent solutions GmbH, Ulm; 2006.

[Rybak02] Rybak, H.; *PDM - was hat sich verändert;* Produktdatenmanagement-Magazin, CAD-CAM REPORT No. 7, Munich; 2002.

[SAP01] SAP; *"Managing the E-Supply Chain" in der Automobilindustrie mit mySAP Automotive;* White Paper of SAP AG, Walldorf; 2001.

[SapPlm01] SAP PLM-Team; *mySAP Product Lifecycle Management – Strategie, Technologie, Implementierung;* SAP Press/Galileo Press; 2001.

[Schleg78] Schlegel, H.; *Betriebswirtschaftliche Konsequenzen der Produktdifferenzierung dargestellt am Beispiel der Variantenvielfalt im Automobilbau;* In: WiSt - Economics Studies 7, Vol. 2, pp. 65-83; 1978.

[Schöns85] Schönsleben, P.; *Flexible Produktionsplanung und -steuerung mit dem Computer: Erweiterungen der Datenbank und Einsatz eines Expertensystems für eine variantenreiche Fertigung;* CW Publications, Munich; 1985.

[Schött99] Schöttner, Josef; *Produktdatenmanagement in der Fertigungsindustrie: Prinzip - Konzepte - Strategien;* Carl Hanser, Munich/Vienna; 1999.

[Schuh89] Schuh, G.; *Gestaltung und Bewertung von Produktvarianten - Ein Beitrag zur systematischen Planung von Serienprodukten;* VDI Research Reports, Vol. 2, No. 177, VDI, Düsseldorf; 1989.

[Seybol97] Seybold, E. et al.; *NED-WORK - die reine Form der Produktbeschreibung;* Internal paper, DaimlerChrysler AG; 1997.

[Sheikh03] Sheikh, A.; *Vergleich zweier Datenmodelle im Bereich der KFZ-Produktion in Hinblick auf ihre methodischen Einsatzmöglichkeiten;* Diploma thesis at Bremerhaven University, Berlin; 2003.

[Specht94] Specht, O.; Ahrens, D.; Wolter, B.; *Material- und Fertigungswirtschaft: Produktionslogistik mit PPS-Systemen;* Friedrich Kiehl, Ludwigshafen; 1994.

[Steinm99] Steinmeier, E.; *Realisierung eines systematischen Produktmodells;* Dissertation at Technical University Munich, Shaker Verlag, Aachen; 1999.

[Thaler01] Thaler, Klaus; *Supply Chain Management - Prozessoptimierung in der logistischen Kette;* Fortis Verlag, Cologne; 2001.

[Thieße96] Thießen, F.; *Erfolgswirkungen der Variantenvielfalt und Variantenmanagement: Anmerkungen zum Beitrag von Adolf G. Coenenberg und Martin Prillmann;* In: ZfB - Magazine for Business Management 66, Vol. 8, pp. 989-992; 1996.

[Trippn02] Trippner, Dietmar; *Vorgehensmodell zum Management von Produktdaten in komplexen und dynamischen Produktentwicklungsprozessen;* Shaker, Aachen; 2002.

[Ungehe86] Ungeheuer, U.; *Produkt- und Montagestrukturierung: Methodik zur Planung einer anforderungsgerechten Produkt- und Montagestruktur für komplexe Erzeugnisse der Einzel- und Kleinserienproduktion;* VDI Progress Report, Vol. 2, No. 112, VDI, Düsseldorf; 1986.

[VDIBer_97] Verein Deutscher Ingenieure e.V. (VDI) (ed.); *Neue Generation von CAD/CAM-Systemen – Erfüllte und enttäuschte Erwartungen;* VDI Report Vol. 1357, VDI, Düsseldorf; 1997.

[VDIT28_72] Verein Deutscher Ingenieure e.V. (VDI) (ed.); *Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung III: Informations- und Stücklistenwesen;* VDI Pocket Book T28, VDI, Düsseldorf; 1972.

[VDIT77_76] Verein Deutscher Ingenieure e.V. (VDI) (ed.); *Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung VI: Begriffszusammenhänge, Begriffsdefinitionen;* VDI Pocket Book T77, VDI, Düsseldorf; 1976.

[Volksw96] Volkswagen (ed.); *Golf Cabriolet. Price list: Effective for Model Year 1997, As of 1996-03-27;* Volkswagen AG, Wolfsburg; 1996.

[Wahl95] Wahl, A.H.; *Optimale Vertriebsplanung und -steuerung im Primärbedarf mit Hilfe der linearen Optimierung;* In: G. Bol, T. Christ, B. Suchanek (ed.): Das Potential der Lean-Techniken in der Kraftfahrzeugwirtschaft: Aktuelle Entwicklungen der Material-Logistik in Theorie und Praxis. Materials on Automotive Industry, VDA series Vol. 6, pp. 153-161; 1995.

[Warnec93] Warnecke, H.-J.; *Revolution in der Unternehmenskultur - Das fraktale Unternehmen 2nd edition (1st ed.: Die fraktale Fabrik);* Springer, Berlin; 1993.

[WiCase] Wikipedia Encyclopedia; *Computer Aided Software Engineering-Tools,* July 5, 2006, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Computer-Aided_Software_Engineering

[Wienda95] Wiendahl, H.-P.; *Produktionsplanung und -steuerung im Wandel;* In: ZwF - Magazine for Economic Manufacturing 90, Vol. 3, pp. 82-85; 1995.

[Wijnke03] Wijnker, T.C.; *Integration of Information in Manufacturing Systems;* Dissertation at Twente University, Netherlands; 2003.

[WiMVC] Wikipedia Encyclopedia; *Model-View-Controller-Paradigma,* July 4, 2006, Wikimedia Foundation Inc., http://en.wikipedia.org/wiki/Model-view-controller

[Womack91] Womack, J.P.; Jones, D.T.; Roos, D.; *Die zweite Revolution in der Automobilindustrie – Konsequenzen aus der weltweiten Studie des Massachusetts Institute of Technology;* Campus, Frankfurt/New York; 1991.

[WWWm1] www.merriam-webster.com

[WWWp1] http://protege.stanford.edu

[WWWs1] http://java.sun.com

[WWWu1] http://www.oose.de/downloads/uml-2-Notationsuebersicht-oose.de.pdf

[Zäpfel91] Zäpfel, G.; *Stücklisten, Verwendungsnachweise, Arbeitspläne und Produktionsfunktionen;* In: WiSt - Economics Studies, 20, Vol. 7, pp. 340-346; 1991.

[Zieman84] Ziemann, W.; *Teilebedarfsermittlung bei auftragsorientierter Wiederholfertigung und Lieferung auf Abruf, dargestellt am Beispiel des Aufzugbaus;* Dissertation at the Faculty for Mechanical and Electrical Engineering at Braunschweig Technical University, Braunschweig; 1984.

Index

M

N

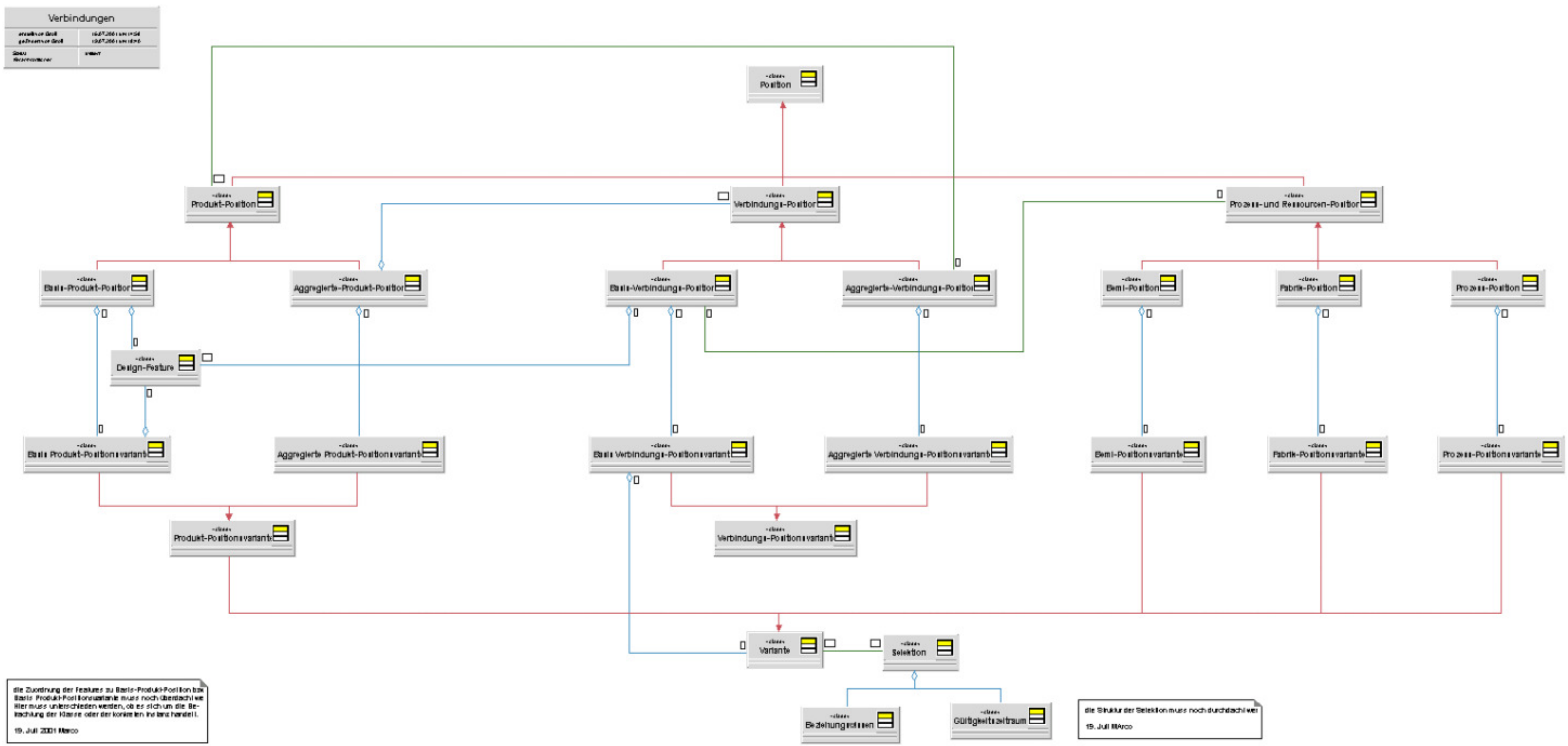
O

P

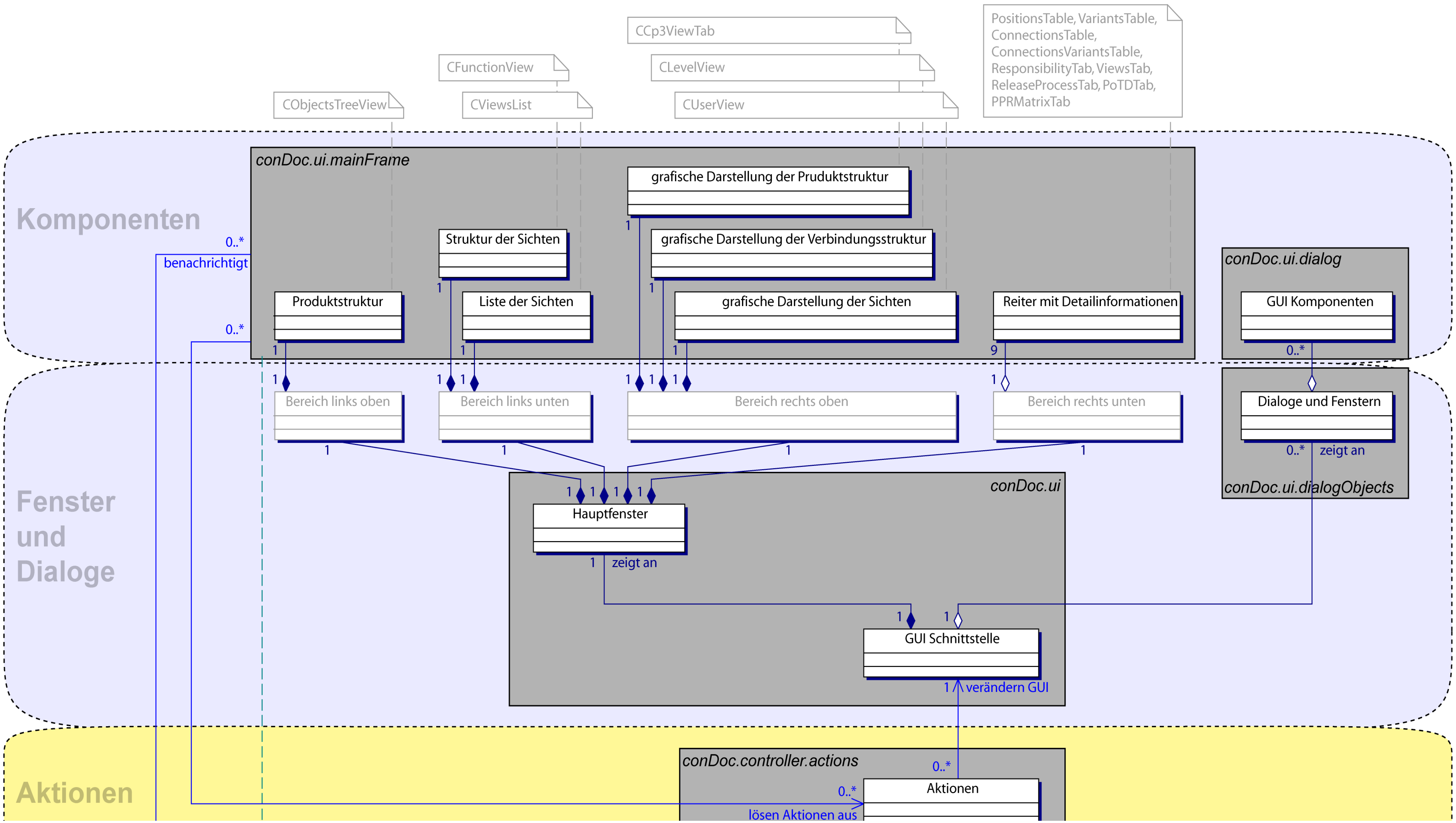
R

Appendix

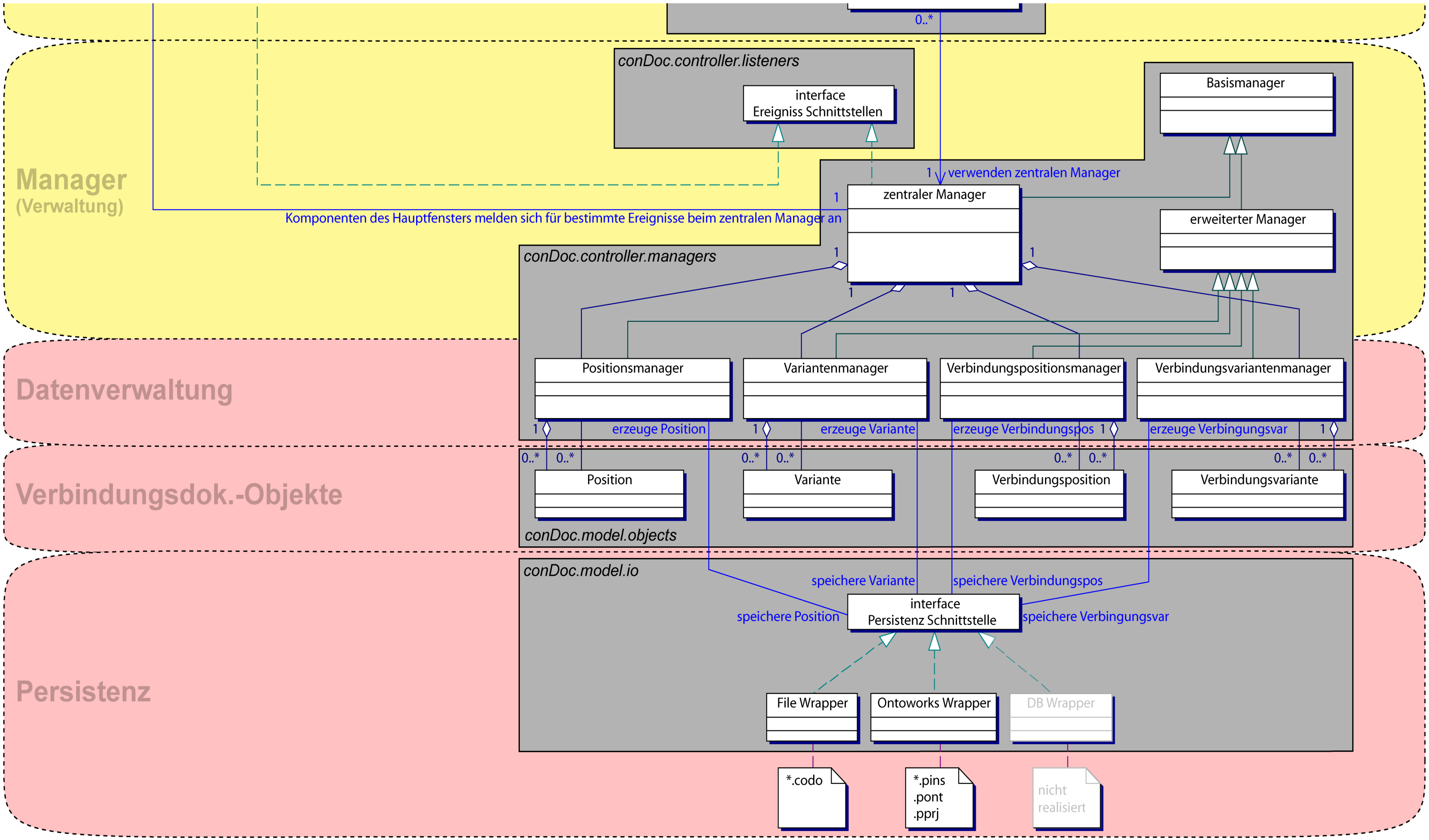
A UML diagrams



Appendix A: Class diagram of the interconnections documentation



CObjectsTreeView
CViewsList
CFunctionView
CCp3ViewTab
CLevelView
CUserView
PositionsTable, VariantsTable, ConnectionsTable, ConnectionsVariantsTable, ResponsibilityTab, ViewsTab, ReleaseProcessTab, PoTDTab, PPRMatrixTab
Komponenten
conDoc.ui.mainFrame
grafische Darstellung der Pruduktstruktur
Struktur der Sichten
grafische Darstellung der Verbindungsstruktur
Produktstruktur
Liste der Sichten
grafische Darstellung der Sichten
Reiter mit Detailinformationen
conDoc.ui.dialog
GUI Komponenten
0..*
benachrichtigt
Fenster und Dialoge
Bereich links oben
Bereich links unten
Bereich rechts oben
Bereich rechts unten
Dialoge und Fenstern
zeigt an
conDoc.ui.dialogObjects
conDoc.ui
Hauptfenster
GUI Schnittstelle
verändern GUI
Aktionen
conDoc.controller.actions
lösen Aktionen aus
1
9



Appendix A: Class diagram of the system architecture of the prototype

B "ITego" data

Appendix B: Product data of "ITego"
(Source [Maximi07])

System	Navigation Bucket	Bucket	Position	Bucket Name	Variant	Code Rule	Part Number	Version	Validity Period	CAD Model	Port 1	Port 2	Port 3	Port 4	...	Attribute 1	Attribute ...	Customer Order 1	Customer Order ...
Chassis	MC Axles	400	010	Front Axle	001	;	HXY0010010102_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Chassis	MC Axles	2XP	020	Front Axle Tandem	001	;	HXY0010010102_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Chassis	MC Axles	400	030	Rear Axle	001	;	HXY0010010102_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Chassis	Cab Suspension System	650	040	Cab Mounting, complete	001	;	HXY0010020110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Cab Suspension System	683	050	Sleepercab mounting	001	;	HXY0010020111_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Exhaust System	1WV	060	Connecting Exhaust Pipes	001	;	HXY0010030107_1	001	01.01.2006 - 31.12.2006		n1	h1	h2	s1	s2				
Chassis	Exhaust System	1WV	070	Connecting Exhaust Pipes	001	;	HXY0010030107_1	001	01.01.2006 - 31.12.2006		n1	h1	h2	s1	s2				
Chassis	Exhaust System	236	080	Front Exhaust Pipe	001	;	HXY0010050104_1	001	01.01.2006 - 31.12.2006		d1								
Chassis	Exhaust System	236	090	Front Exhaust Pipe	001	;	HXY0010050104_1	001	01.01.2006 - 31.12.2006		d1								
Chassis	Chassis Mounted Equipme	2Z2	100	License Plate Mounting com	001	;	HXY0010010103_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	s1				
Chassis	Chassis Mounted Equipme	09M	110	Fifth Wheel Complete	001	;	HXY0010010104_1	001	01.01.2006 - 31.12.2006		h1	h2	h3	h4					
Chassis	Steps&Chassis Fairings	67B	120	Cab Access steps complete	001	;	HXY0010020101_1	001	01.01.2006 - 31.12.2006		n1	h1	s1	s2	qh1				
Chassis	Steps&Chassis Fairings	67B	130	Cab Access steps complete	001	;	HXY0010020101_1	001	01.01.2006 - 31.12.2006		n1	h1	s1	s2	qh1				
Chassis	Steps&Chassis Fairings	67B	140	Cab Access steps complete	001	;	HXY0010020102_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	s1				
Chassis	Steps&Chassis Fairings	67B	150	Cab Access steps complete	001	;	HXY0010020103_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Chassis	Steps&Chassis Fairings	10Q	160	Chassis mounted steps	001	;	HXY0010020102_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	s1				
Chassis	Steps&Chassis Fairings	10Q	170	Chassis mounted steps	001	;	HXY0010020103_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Chassis	Fuel System	115	180	Fuel tank(s) complete	001	;	HXY0010020105_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	190	Fuel tank(s) complete	001	;	HXY0010020105_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	200	Fuel tank(s) complete	001	;	HXY0010020104_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	s1					

Chassis	Fuel System	115	210	Fuel tank(s) complete	001	;	HXY0010020108_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	220	Fuel tank(s) complete	001	;	HXY0010020106_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	230	Fuel tank(s) complete	001	;	HXY0010020107_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	240	Fuel tank(s) complete	001	;	HXY0010020105_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	250	Fuel tank(s) complete	001	;	HXY0010020105_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	260	Fuel tank(s) complete	001	;	HXY0010020104_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	s1					
Chassis	Fuel System	115	270	Fuel tank(s) complete	001	;	HXY0010020108_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	280	Fuel tank(s) complete	001	;	HXY0010020106_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Fuel System	115	290	Fuel tank(s) complete	001	;	HXY0010020107_1	001	01.01.2006 - 31.12.2006		n1	h1							
Chassis	Tires & Wheels	13P	300	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13P	310	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13P	320	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13P	330	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13P	340	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13P	350	Tire complete	001	;	HXY0010050106_1	001	01.01.2006 - 31.12.2006		r1								
Chassis	Tires & Wheels	13Q	360	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Tires & Wheels	13Q	370	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Tires & Wheels	13Q	380	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Tires & Wheels	13Q	390	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Tires & Wheels	13Q	400	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Tires & Wheels	13Q	410	Wheels complete	001	;	HXY0010050107_1	001	01.01.2006 - 31.12.2006		kl1	d1							
Chassis	Frame	09A	420	Frame Rail - complete	001	;	HXY0010010101_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Frame	09A	430	Frame Rail - complete	001	;	HXY0010010101_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Frame	562	440	Midship No 1 Crossmember	001	;	HXY0010010106_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Frame	563	450	Midship No 2 Crossmember	001	;	HXY0010010107_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Chassis	Frame	564	460	Midship No 3 Crossmember	001	;	HXY0010010108_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Chassis	Frame	559	470	Engine crossmember	001	;	HXY0010020109_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Aerodynamic Devices	006	480	Aerodynamic Device - Roof	001	;	HXY0010050101_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Cab/Body	Aerodynamic Devices	006	490	Aerodynamic Device - Roof	001	;	HXY0010050102_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	820	500	Cab floor - complete	001	;	HXY0010030103_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	705	510	Cab - Additional Requireme	001	;	HXY0010030106_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	520	Sleepercab/sleeperbox skin	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	530	Sleepercab/sleeperbox skin	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	01D	540	Cab sidewall - complete	001	;	HXY0010030113_1	001	01.01.2006 - 31.12.2006		h1	h2							
Cab/Body	Cab in white	01D	550	Cab sidewall - complete	001	;	HXY0010030113_1	001	01.01.2006 - 31.12.2006		h1	h2							
Cab/Body	Cab in white	824	560	Cab front wall - complete	001	;	HXY0010030112_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	823	570	Cab back wall - complete	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	580	Sleepercab/sleeperbox skin	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	590	Sleepercab/sleeperbox skin	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				

Cab/Body	Cab in white	682	600	Sleepercab/sleeperbox skin/	001	;	HXY0010040107_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	610	Sleepercab/sleeperbox skin/	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	682	620	Sleepercab/sleeperbox skin/	001	;	HXY0010030110_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Cab in white	825	630	Cab roof - complete	001	;	HXY0010040108_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Cab/Body	Cab in white	682	640	Sleepercab/sleeperbox skin/	001	;	HXY0010040111_1	001	01.01.2006 - 31.12.2006		n1	h1	h2						
Cab/Body	Cab in white	682	650	Sleepercab/sleeperbox skin/	001	;	HXY0010040111_1	001	01.01.2006 - 31.12.2006		n1	h1	h2						
Cab/Body	Cab in white	825	660	Cab roof - complete	001	;	HXY0010040114_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Cab/Body	Cab in white	825	670	Cab roof - complete	001	;	HXY0010050103_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2					
Cab/Body	Dash	680	680	Dash - complete	001	;	HXY0010040104_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	s1				
Cab/Body	Door System	651	690	Front door left side	001	;	HXY0010040105_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Door System	653	700	Front door right side	001	;	HXY0010040106_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Exterior Equipment & Acce	82A	710	Cab/Body Trim exterior	001	;	HXY0010030101_1	001	01.01.2006 - 31.12.2006		h1	h2	h3	h4					
Cab/Body	Exterior Equipment & Acce	2CW	720	Product Names & logos	001	;	HXY00100_sticker	001	01.01.2006 - 31.12.2006										
Cab/Body	Exterior Equipment & Acce	2E3	730	Grabhandels - exterior	001	;	HXY0010040112_1	001	01.01.2006 - 31.12.2006		n1	h1	r1						
Cab/Body	Exterior Equipment & Acce	2E3	740	Grabhandels - exterior	001	;	HXY0010040112_1	001	01.01.2006 - 31.12.2006		n1	h1	r1						
Cab/Body	Exterior Equipment & Acce	2E3	750	Grabhandels - exterior	001	;	HXY0010040113_1	001	01.01.2006 - 31.12.2006		d1								
Cab/Body	Exterior Equipment & Acce	2E3	760	Grabhandels - exterior	001	;	HXY0010040113_1	001	01.01.2006 - 31.12.2006		d1								
Cab/Body	Exterior Equipment & Acce	82A	770	Cab/Body Trim exterior	001	;	HXY0010040115_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Cab/Body	Exterior Equipment & Acce	82A	780	Cab/Body Trim exterior	001	;	HXY0010040115_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Cab/Body	Fenders, Bumpers, Hood &	556	790	Front Bumper - complete	001	;	HXY0010010105_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	h3				
Cab/Body	Fenders, Bumpers, Hood &	667	800	Front Fender - complete	001	;	HXY0010030102_2	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				
Cab/Body	Fenders, Bumpers, Hood &	643	810	Hood Hardware	001	;	HXY0010030105_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	h1				
Cab/Body	Fenders, Bumpers, Hood &	646	820	Grille	001	;	HXY0010020102_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2					
Cab/Body	Fenders, Bumpers, Hood &	644	830	Hood Engine - enclosure	001	;	HXY0010040101_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	h3				
Cab/Body	Fenders, Bumpers, Hood &	649	840	Grille Guard / Front Cab Gu	001	;	HXY0010040103_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Cab/Body	Fenders, Bumpers, Hood &	03L	850	Hood Complete	001	;	HXY0010040109_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2	h3				
Cab/Body	Pedals & Interior Steering	532	860	Steering Column - complete	001	;	HXY0010030111_1	001	01.01.2006 - 31.12.2006		h1	h2							
Cab/Body	Lamps	300	870	Turn signal lamp - front	001	;	HXY0010030109_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Lamps	300	880	Turn signal lamp - front	001	;	HXY0010030109_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Lamps	312	890	Headlamp - complete	001	;	HXY0010040102_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Lamps	312	900	Headlamp - complete	001	;	HXY0010040102_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Audible Warnings	727	910	Horn air - complete	001	;	HXY0010040112_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Audible Warnings	727	920	Horn air - complete	001	;	HXY0010040112_1	001	01.01.2006 - 31.12.2006		n1	h1							
Cab/Body	Audible Warnings	727	930	Horn air - complete	001	;	HXY0010050105_1	001	01.01.2006 - 31.12.2006		d1								
Cab/Body	Audible Warnings	727	940	Horn air - complete	001	;	HXY0010050105_1	001	01.01.2006 - 31.12.2006		d1								
Cab/Body	Glass & Sealing	663	950	Windshield - complete	001	;	HXY0010040110_1	001	01.01.2006 - 31.12.2006		n1	n2	h1	h2					
Cab/Body	Seats/Seatbelts	1S7	960	Front seat - constructed	001	;	HXY0010030104_1	001	01.01.2006 - 31.12.2006		h1	h2	s1	s2	s3				
Powertrain	MC Powerpack	101	970	Major component engine	001	;	HXY0010030108_1	001	01.01.2006 - 31.12.2006		n1	n2	n3	n4	n5				

Appendix B: Process data of "ITego"
(Source [Maximi07])

Position	Name	Variant	Code Rule	Connection 1							Connection 2							Version	Validity Period	Attribute 1	Attribute ...	Customer Order 1	Customer Order ...
				Position	Variant	Code Rule	Part Number with Version	Port 1	Port 2	Port 3	Position	Variant	Code Rule	Part Number with Version	Port 1	Port 2	Port 3						
010	Frame rail - complete	001	;	420	001	;	HXY00100101	s1			430	001	;	HXY0010010	s3			001	01.01.2006 -				
020	License plate mounting -	001	;	100	001	;	HXY00100101	s1.1			420	001	;	HXY0010010	s2			001	01.01.2006 -				
030	License plate mounting -	001	;	100	001	;	HXY00100101	s1.2			430	001	;	HXY0010010	s2			001	01.01.2006 -				
040	Front bumper complete	001	;	790	001	;	HXY00100101	h1			420	001	;	HXY0010010	n1			001	01.01.2006 -				
050	Front bumper complete	001	;	790	001	;	HXY00100101	h2			430	001	;	HXY0010010	n2			001	01.01.2006 -				
060	Fifth wheel complete	001	;	110	001	;	HXY00100101	h1			420	001	;	HXY0010010	n16			001	01.01.2006 -				
070	Fifth wheel complete	001	;	110	001	;	HXY00100101	h2			430	001	;	HXY0010010	n16			001	01.01.2006 -				
080	Fifth wheel complete	001	;	110	001	;	HXY00100101	h3	h4		100	001	;	HXY0010010	n1	n2		001	01.01.2006 -				
090	Front Axle	001	;	010	001	;	HXY00100101	n1			420	001	;	HXY0010010	h2			001	01.01.2006 -				
100	Front Axle	001	;	010	001	;	HXY00100101	n2			430	001	;	HXY0010010	h2			001	01.01.2006 -				
110	Rear Axle Tandem	001	;	020	001	;	HXY00100101	n1			420	001	;	HXY0010010	h12			001	01.01.2006 -				
120	Rear axle Tandem	001	;	020	001	;	HXY00100101	n2			430	001	;	HXY0010010	h12			001	01.01.2006 -				
130	Rear axle	001	;	030	001	;	HXY00100101	n1			420	001	;	HXY0010010	h15			001	01.01.2006 -				
140	Rear axle	001	;	030	001	;	HXY00100101	n2			430	001	;	HXY0010010	h15			001	01.01.2006 -				
150	Midship No1 Crossmemb	001	;	440	001	;	HXY00100101	n2	n3		420	001	;	HXY0010010	h5	h6		001	01.01.2006 -				
160	Midship No1 Crossmemb	001	;	440	001	;	HXY00100101	n6	n7		430	001	;	HXY0010010	h5	h6		001	01.01.2006 -				
170	Midship No2 Crossmemb	001	;	450	001	;	HXY00100101	n2			420	001	;	HXY0010010	h9			001	01.01.2006 -				
180	Midship No2 Crossmemb	001	;	450	001	;	HXY00100101	n3			430	001	;	HXY0010010	h9			001	01.01.2006 -				
190	Midship No3 Crossmemb	001	;	460	001	;	HXY00100101	n1	n2	n3	420	001	;	HXY0010010	h10	h11	h12	001	01.01.2006 -				
200	Midship No3 Crossmemb	001	;	460	001	;	HXY00100101	n5	n6	n7	430	001	;	HXY0010010	h10	h11	h12	001	01.01.2006 -				
210	Cab Access Steps compl	001	;	120	001	;	HXY00100201	s1			130	001	;	HXY0010020	s1			001	01.01.2006 -				
220	Cab Access Steps compl	001	;	140	001	;	HXY00100201	h1	h2		130	001	;	HXY0010020	n1			001	01.01.2006 -				
230	Cab Access Steps compl	001	;	140	001	;	HXY00100201	h1	h2		130	001	;	HXY0010020	n1			001	01.01.2006 -				
240	Cab Access Steps compl	001	;	150	001	;	HXY00100201	h1	h2		140	001	;	HXY0010020	n1	n2		001	01.01.2006 -				
250	Cab Access Steps compl	001	;	120	001	;	HXY00100201	h1			440	001	;	HXY0010010	n1			001	01.01.2006 -				
260	Cab Access Steps compl	001	;	130	001	;	HXY00100201	h1			440	001	;	HXY0010010	n5			001	01.01.2006 -				

270	Cab Access Steps compl	001	;	120	001	;	HXY00100201	h1			440	001	;	HXY0010010	n4			001	01.01.2006 -				
280	Cab Access Steps compl	001	;	130	001	;	HXY00100201	h1			440	001	;	HXY0010010	n8			001	01.01.2006 -				
290	Engine Crossmember	001	;	470	001	;	HXY00100201	h1	h3		420	001	;	HXY0010010	n2	n3		001	01.01.2006 -				
300	Engine Crossmember	001	;	470	001	;	HXY00100201	h2	h4		430	001	;	HXY0010010	n2	n3		001	01.01.2006 -				
310	Fuel Tanks complete	001	;	240	001	;	HXY00100201	h1			260	001	;	HXY0010020	n1			001	01.01.2006 -				
320	Fuel Tanks complete	001	;	250	001	;	HXY00100201	h1			260	001	;	HXY0010020	n1			001	01.01.2006 -				
330	Fuel Tanks complete	001	;	200	001	;	HXY00100201	n3			270	001	;	HXY0010020	h1			001	01.01.2006 -				
340	Fuel Tanks complete	001	;	200	001	;	HXY00100201	n2			280	001	;	HXY0010020	h1			001	01.01.2006 -				
350	Fuel Tanks complete	001	;	270	001	;	HXY00100201	n1			290	001	;	HXY0010020	h1			001	01.01.2006 -				
360	Fuel Tanks complete	001	;	250	001	;	HXY00100201	h1			450	001	;	HXY0010010	n1			001	01.01.2006 -				
370	Chassis mounted steps	001	;	160	001	;	HXY00100201	h1			420	001	;	HXY0010010	n12			001	01.01.2006 -				
380	Chassis mounted steps	001	;	170	001	;	HXY00100201	h2			430	001	;	HXY0010010	n12			001	01.01.2006 -				
390	Fuel Tanks complete	001	;	240	001	;	HXY00100201	h1			200	001	;	HXY0010020	n1			001	01.01.2006 -				
400	Fuel Tanks complete	001	;	250	001	;	HXY00100201	h1			200	001	;	HXY0010020	n1			001	01.01.2006 -				
410	Fuel Tanks complete	001	;	260	001	;	HXY00100201	n2			270	001	;	HXY0010020	h1			001	01.01.2006 -				
420	Fuel Tanks complete	001	;	260	001	;	HXY00100201	n3			280	001	;	HXY0010020	h1			001	01.01.2006 -				
430	Fuel Tanks complete	001	;	270	001	;	HXY00100201	n1			290	001	;	HXY0010020	h1			001	01.01.2006 -				
440	Fuel Tanks complete	001	;	190	001	;	HXY00100201	h1			290	001	;	HXY0010010	n4			001	01.01.2006 -				
450	Cab mounting complete	001	;	040	001	;	HXY00100201	h2	h6	h10	420	001	;	HXY0010010	n4	n5	n6	001	01.01.2006 -				
460	Cab mounting complete	001	;	040	001	;	HXY00100201	h3	h7	h11	430	001	;	HXY0010010	n4	n5	n6	001	01.01.2006 -				
470	Sleepercab Mounting	001	;	050	001	;	HXY00100201	h3	h9	h15	420	001	;	HXY0010010	n8	n9	n10	001	01.01.2006 -				
480	Sleepercab Mounting	001	;	050	001	;	HXY00100201	h4	h10	h16	430	001	;	HXY0010010	n8	n9	n10	001	01.01.2006 -				
490	Cab/body trim exterior	001	;	710	001	;	HXY00100301	h1	h2	h3	790	001	;	HXY0010010	n1	n2	n3	001	01.01.2006 -				
500	Front Fender Complete	001	;	800	001	;	HXY00100301	h2	h3	h6	470	001	;	HXY0010020	n1	n2	n3	001	01.01.2006 -				
510	Cab Floor Complete	001	;	500	001	;	HXY00100301	h1	h2	h3	040	001	;	HXY0010020	n2	n3	n6	001	01.01.2006 -				
520	Front seat constructed	001	;	960	001	;	HXY00100301	h1	h2		040	001	;	HXY0010020	n10	n11		001	01.01.2006 -				
530	Chassis mounted steps	001	;	160	001	;	HXY00100201	h1	h2		170	001	;	HXY0010020	n1	n2		001	01.01.2006 -				
540	Hood Hardware	001	;	810	001	;	HXY00100301	h1	h2	h3	790	001	;	HXY0010010	n1	n2		001	01.01.2006 -				
550	Connecting Exhaust pipe	001	;	060	001	;	HXY00100301	h1			040	001	;	HXY0010020	n13			001	01.01.2006 -				
560	Connecting Exhaust pipe	001	;	070	001	;	HXY00100301	h1			040	001	;	HXY0010020	n16			001	01.01.2006 -				
570	Cab additional requireme	001	;	510	001	;	HXY00100301	h1	h2	h3	040	001	;	HXY0010020	n14	n15		001	01.01.2006 -				
580	Major component engine	001	;	970	001	;	HXY00100301	h1	h2	h3	500	001	;	HXY0010030	n1	n2		001	01.01.2006 -				
590	Turn signal lamp front	001	;	870	001	;	HXY00100301	h1			810	001	;	HXY0010030	n1			001	01.01.2006 -				
600	Turn signal lamp front	001	;	880	001	;	HXY00100301	h1			810	001	;	HXY0010030	n4			001	01.01.2006 -				
610	sleepercab/sleeperbox sk	001	;	580	001	;	HXY00100301	h1	h2	h3	050	001	;	HXY0010020	n1	n2	n3	001	01.01.2006 -				
620	sleepercab/sleeperbox sk	001	;	590	001	;	HXY00100301	h1	h2	h3	050	001	;	HXY0010020	n1	n2	n3	001	01.01.2006 -				
630	cab side wall complete	001	;	550	001	;	HXY00100301	h1	h2		870	001	;	HXY0010030	n1			001	01.01.2006 -				

640	cab side wall complete	001	;	550	001	;	HXY0010030	h1	h2		800	001	;	HXY0010030	n1			001	01.01.2006 -				
650	cab side wall complete	001	;	550	001	;	HXY0010030	h1	h2		880	001	;	HXY0010030	n1			001	01.01.2006 -				
660	cab side wall complete	001	;	550	001	;	HXY0010030	h1	h2		800	001	;	HXY0010030	n4			001	01.01.2006 -				
670	cab front wall complete	001	;	560	001	;	HXY0010030	h1	h2		800	001	;	HXY0010030	n5	n8		001	01.01.2006 -				
680	cab front wall complete	001	;	560	001	;	HXY0010030	h3	h4		970	001	;	HXY0010030	n5	n6		001	01.01.2006 -				
690	steering column - comple	001	;	860	001	;	HXY0010030	h1	h2		970	001	;	HXY0010030	n7	n8		001	01.01.2006 -				
700	cab back wall complete	001	;	570	001	;	HXY0010030	h1			060	001	;	HXY0010030	n1			001	01.01.2006 -				
710	cab back wall complete	001	;	570	001	;	HXY0010030	h4			070	001	;	HXY0010030	n1			001	01.01.2006 -				
720	sleepercab/sleeperbox sk	001	;	580	001	;	HXY0010030	h1	h2	h3	640	001	;	HXY0010030	n1	n2	n3	001	01.01.2006 -				
730	sleepercab/sleeperbox sk	001	;	590	001	;	HXY0010030	h1	h2	h3	650	001	;	HXY0010030	n1	n2	n3	001	01.01.2006 -				
740	grille	001	;	820	001	;	HXY0010020	h1	h2		830	001	;	HXY0010040	n4	n5		001	01.01.2006 -				
750	headlamp complete	001	;	890	001	;	HXY0010040	h1			830	001	;	HXY0010040	n3			001	01.01.2006 -				
760	headlamp complete	001	;	900	001	;	HXY0010040	h1			830	001	;	HXY0010040	n6			001	01.01.2006 -				
770	grille guard / front cab gu	001	;	840	001	;	HXY0010040	h1	h2		820	001	;	HXY0010020	n1	n2		001	01.01.2006 -				
780	hood engine enclosure	001	;	830	001	;	HXY0010040	h1	h2		970	001	;	HXY0010030	n1	n2		001	01.01.2006 -				
790	sleepercab/sleeperbox sk	001	;	600	001	;	HXY0010040	h1	h2		670	001	;	HXY0010030	n2	n3		001	01.01.2006 -				
800	product names & logos	001	;	720	001	;	HXY00100_st	h2	h3		050	001	;	HXY0010020	n20	n21	n22	001	01.01.2006 -				
810	dash complete	001	;	680	001	;	HXY0010040	h1	h2		560	001	;	HXY0010030	n2	n3		001	01.01.2006 -				
820	Front door left side	001	;	690	001	;	HXY0010040	h1			040	001	;	HXY0010020	n1			001	01.01.2006 -				
830	Front door right side	001	;	700	001	;	HXY0010040	h1			040	001	;	HXY0010020	n4			001	01.01.2006 -				
840	sleepercab/sleeperbox sk	001	;	650	001	;	HXY0010030	h1	h2	h3	650	001	;	HXY0010030	n1	n2	n3	001	01.01.2006 -				
850	sleepercab/sleeperbox sk	001	;	650	001	;	HXY0010030	h1	h2	h3	650	001	;	HXY0010030	n1	n2	n3	001	01.01.2006 -				
860	Hood complete	001	;	850	001	;	HXY0010040	h3	h4		830	001	;	HXY0010040	n1	n2		001	01.01.2006 -				
870	cab roof complete	001	;	670	001	;	HXY0010040	h1	h7		580	001	;	HXY0010030	n3	n4		001	01.01.2006 -				
880	cab roof complete	001	;	670	001	;	HXY0010040	h6	h12		590	001	;	HXY0010030	n3	n4		001	01.01.2006 -				
890	windshield complete	001	;	950	001	;	HXY0010040	h1	h4		850	001	;	HXY0010040	n1	n2		001	01.01.2006 -				
900	sleepercab/sleeperbox sk	001	;	640	001	;	HXY0010040	h1	h2		640	001	;	HXY0010040	n1	n2		001	01.01.2006 -				
910	sleepercab/sleeperbox sk	001	;	650	001	;	HXY0010040	h1	h2		650	001	;	HXY0010040	n1	n2		001	01.01.2006 -				
920	grabhandles exterior	001	;	730	001	;	HXY0010040	h1			630	001	;	HXY0010040	n7			001	01.01.2006 -				
930	grabhandles exterior	001	;	740	001	;	HXY0010040	h1			630	001	;	HXY0010040	n12			001	01.01.2006 -				
940	grabhandles exterior	001	;	750	001	;	HXY0010040	d1			730	001	;	HXY0010040	r1			001	01.01.2006 -				
950	grabhandles exterior	001	;	760	001	;	HXY0010040	d1			740	001	;	HXY0010040	r1			001	01.01.2006 -				
960	cab roof complete	001	;	660	001	;	HXY0010040	h1	h2	h3	630	001	;	HXY0010040	n2	n3	n4	001	01.01.2006 -				
970	cab body trim exterior	001	;	770	001	;	HXY0010040	h1			640	001	;	HXY0010040	n1			001	01.01.2006 -				
980	cab body trim exterior	001	;	780	001	;	HXY0010040	h2			630	001	;	HXY0010040	n6			001	01.01.2006 -				
990	cab body trim exterior	001	;	770	001	;	HXY0010040	h1			640	001	;	HXY0010040	n1			001	01.01.2006 -				
1000	cab body trim exterior	001	;	780	001	;	HXY0010040	h2			630	001	;	HXY0010040	n1			001	01.01.2006 -				

1010	Aerodynamic device roof	001	;	480	001	;	HXY00100501	h1	h2	h3	490	001	;	HXY0010050	n1	n2	n3	001	01.01.2006 -				
1020	Aerodynamic device roof	001	;	480	001	;	HXY00100501	h1	h2		950	001	;	HXY0010040	n1	n2	n3	001	01.01.2006 -				
1030	Aerodynamic device roof	001	;	490	001	;	HXY00100501	h1	h2	h3	660	001	;	HXY0010040	n1	n2	n3	001	01.01.2006 -				
1040	cab roof complete	001	;	670	001	;	HXY00100501	h1	h2		660	001	;	HXY0010040	n6	n7		001	01.01.2006 -				
1050	front exhaust pipe	001	;	080	001	;	HXY00100501	n1			060	001	;	HXY0010030	h1			001	01.01.2006 -				
1060	front exhaust pipe	001	;	090	001	;	HXY00100501	n1			070	001	;	HXY0010030	h1			001	01.01.2006 -				
1070	horn air , complete	001	;	910	001	;	HXY00100401	h1			660	001	;	HXY0010040	n5			001	01.01.2006 -				
1080	horn air , complete	001	;	920	001	;	HXY00100401	h1			660	001	;	HXY0010040	n8			001	01.01.2006 -				
1090	horn air , complete	001	;	930	001	;	HXY00100501	n1			910	001	;	HXY0010040	r1			001	01.01.2006 -				
1100	horn air , complete	001	;	940	001	;	HXY00100501	n1			920	001	;	HXY0010040	r1			001	01.01.2006 -				
1110	tire complete	001	;	300	001	;	HXY00100501	h1			360	001	;	HXY0010050	n1			001	01.01.2006 -				
1120	tire complete	001	;	310	001	;	HXY00100501	h1			370	001	;	HXY0010050	n1			001	01.01.2006 -				
1130	tire complete	001	;	320	001	;	HXY00100501	h1			380	001	;	HXY0010050	n1			001	01.01.2006 -				
1140	tire complete	001	;	330	001	;	HXY00100501	h1			390	001	;	HXY0010050	n1			001	01.01.2006 -				
1150	tire complete	001	;	340	001	;	HXY00100501	h1			400	001	;	HXY0010050	n1			001	01.01.2006 -				
1160	tire complete	001	;	350	001	;	HXY00100501	h1			410	001	;	HXY0010050	n1			001	01.01.2006 -				
1170	wheels complete	001	;	360	001	;	HXY00100501	h1			010	001	;	HXY0010010	n1			001	01.01.2006 -				
1180	wheels complete	001	;	370	001	;	HXY00100501	h1			010	001	;	HXY0010010	n2			001	01.01.2006 -				
1190	wheels complete	001	;	380	001	;	HXY00100501	h1			020	001	;	HXY0010010	n1			001	01.01.2006 -				
1200	wheels complete	001	;	390	001	;	HXY00100501	h1			020	001	;	HXY0010010	n2			001	01.01.2006 -				
1210	wheels complete	001	;	400	001	;	HXY00100501	h1			030	001	;	HXY0010010	n1			001	01.01.2006 -				
1220	wheels complete	001	;	410	001	;	HXY00100501	h1			030	001	;	HXY0010010	n2			001	01.01.2006 -				

"*Only the reduction to the necessary enables completeness.*"

Eckhard Monke