INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

AN INTEGRATED PRODUCT INFORMATION MODEL FOR VARIANT DESIGN IN COMMERCIAL VEHICLE DEVELOPMENT

M. Kreimeyer, C. Baumberger, F. Deubzer and D. Ziethen

Keywords: product architecture, data model, variant management

1. Introduction

With a focus on growing complexity in the markets, a dedicated variant management tackling the whole process from designing new variants all the way to retiring variants from the market gains more and more importance. This is especially true for commercial vehicle design with its highly complex products, small production volumes and a large variety of products in the market. This research was run with a view to implementing a consistent variant design approach in industry, building both a new PDM environment and the necessary processes into an existing design organization.

Compared to the passenger car market the number of manufactured units is relatively small with any commercial vehicle manufacturer, whereas the product variance is significantly higher. Unlike passenger cars, commercial vehicles are a B2B investment good with the specific requirement of catering for the customers' transport tasks most efficiently to improve on their revenue. The overall challenge is to manage this complexity in a market- and customer-oriented way together with a strong emphasis on effective planning of the engineering scope in the early product development phases. This combination puts high demands on collaboration models, product documentation and IT systems.

2. Problem description and research approach

The results presented here summarize the outcome of a longitudinal study of six years at MAN Truck & Bus AG (in short "MAN"). MAN is a large manufacturer of commercial vehicles. The company has production sites in three European countries, with markets and production sites across the globe. Having undergone growth through entering into new markets and through the acquisition of additional companies, the company has been faced – for the last few years – with the needs of managing the ever-growing technical complexity of a growing product portfolio and of a more dedicated design process to target new markets and their new and specific requirements.

Figure 1 shows the initial situation (dashed line) for design projects, as it was perceived by engineering management at the beginning: Based on a rather late start of a design project, only a small percentage of the totally needed variants would be designed, and the properties of the design were only found out about late in the series design process or even during production ramp-up through the testing of physical prototypes. Often, variants would even be added beyond that point, and vehicle designs would occur that could not be sold.

As a conclusion, the target state would, therefore, incorporate an earlier start of design projects (continuous line) starting with a systematic plan for the portfolio to be regarded, and going through four extra stages (labelled 1 through 4 in Figure 1): A planned target portfolio (1), a business case for that

(2), a functional specification for the project based on that (3), and a technical specification that would involve a concept for all variants to be regarded (4).

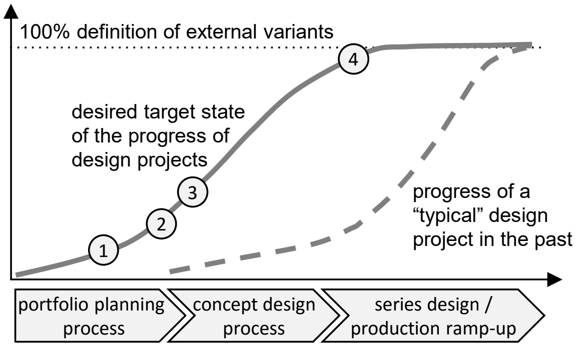


Figure 1. The initial motivation for "Product architecture design"

With this challenge in mind, the initiative was undertaken to rework the overall process design landscape at the company to include an earlier and digitally supported variant design process based on a targeted requirement management, make use of an explicit architectural process, and lay the foundations to digitally manage the design data (esp. bill of material and CAx data). The following goals were, in particular, set to improve the process:

- Establish a generic product structure for the whole organization as a basis for documentation of variants in a manner that components become reusable across different design projects
- Foster "frontloading", i.e. the early assessment of concept designs and their fit to the portfolio as well as their aptitude to cater for the customer requirements early during the design processes while changes are still manageable
- Build a data model that would consistently manage requirements, variant descriptions, the bill of material in the early stages of the design process, and related CAD-data

In order to tackle this challenge "action research" was considered the most appropriate approach to creating a coevolution between science and practice. This coevolution process consisted of frequent iterations including the following:

- Involvement of scientists from the academic area (e.g. bachelor and master projects) as well as domain experts from other industries
- Rapid prototyping by means of demonstrators verified and validated based on real engineering design data together with potential users
- Early support of vehicle design projects and regular reviews with project and line managers

3. State of the art

For an overview of the related state of the art, the definitions available and the solutions that were drawn upon, a short summary of product architecture, product information models and variant design is given. The references were especially used when starting to collect requirements and solution ideas to implement the new function at the company. The material served, in particular, as input to develop the concept for the data model shown below (for a detailed reflection, see [Kreimeyer 2015]).

Considering the state of the art described in the following sections, the initial goals of the research were refined by means of a detailed requirement list. These requirements are summarized in Table 1.

3.1 Product architecture and underlying information models

Literature provides many definitions for product architecture, which have evolved mostly over the past 20 years, starting with [Ulrich 1995] stating that product architecture has three main characteristics:

- the arrangement of functional elements or functional structure
- the mapping of functional elements to physical elements
- the specification of the interfaces among interacting physical elements

In other words, a product architecture consists of both a functional structure and a physical structure. Other authors (for a comprehensive literature review, see esp. [Jose et al. 2005]) have followed this definition, relativizing it in more abstract terms as a configuration between components of the product and the tasks that each component should do or as the "scheme", by which the product's functionality is allocated to the physical structure, which is segmented into the "physical building blocks" and their "interactions" [Ulrich et al. 2012].

The process of designing product architectures is provided by [Schuh et al. 2007] through a procedural model that includes working with requirements, setting up functions and properties of a product, relating the physical components, and the standardization of components, thus making these steps the different domains of "doing architecture". Similarly, the "architecture process" can therefore be seen as converting a desired behaviour (given by requirements and/or functions) to a solution (given by components), but not just for one integral product but a product portfolio, based on modularization to achieve commonality among the different variants within this portfolio.

Especially in the passenger vehicle industry it is common to work with product platforms serving both as a standard and as a basis for several models of the product range that are derived individually from such a platform. "A product platform is a set of architecture standards, common modules and interfaces from which a stream of derivative products can be efficiently developed and launched" [Friedel 2011]. More generally, literature, e.g. [Robertson et al. 1998] defines a platform as the "collection of assets that are shared by a set of products", i.e. they extend the notion on the physical building blocks and their difference and commonality across a range of products, also stating that the standardization of parts alone does not result in a platform. As this definition relates closely to the needs of commercial vehicle portfolios (a mapping of components and functions, incorporating different variants), it is this definition of architecture that is followed here.

3.2 Variant design in product architecture processes

Variant design as such is not directly included in product architecture definitions, but it is rather brought in through the definitions on product platforms: E.g., [Blees 2011] and [Kipp 2011] similarly suggest a procedure to obtain a modular product platform architecture. They integrate the early criteria of the portfolio as part of the requirements collection to designate a variant tree early in the design process. The embodiment of one or more variants in this variant tree is done recruiting the relevant variant drivers (based on [Erixon 1998]) from the whole product lifecycle.

Table 1. Requirements on the solution

REQUIREMENT	REASONING/DESCRIPTION
common product structure	common language for the product; modular decomposition logic; basis for how variants are "cut" and "configured"
one common variant description	one consistent language for authoring and management of functional and geometrical variants based on Boolean description
requirements traceability	ability to later find out why a variant was designed
early bill of material	management of parts and alternative concepts even when no part numbers are available yet
architecture planning	forward planning by means of mapping requirements on technical solutions including geometrical spaces
management of virtual designs	association of all CAD models, related formats (e.g. lightweight data) and processes (e.g. package planning and DMU processes) to the product structure and its variants
consistency with downstream systems	adapt downstream IT systems to be compliant with the new data model
ease of understanding	ensure that future users can easily navigate the data and understand the data model
product portfolio description	manage the market perspective both for future and existing products to enable the design-in-context of new functionality into the current portfolio
change processes for all information	manage the design changes and the maturity of all information

In vehicle industry, the requirements specification process is hardly done systematically. [Almefelt et al. 2006] provides a comprehensive overview based on an extensive empirical study. While relativizing smaller details, they conclude “procedures prescribed for development projects in industry are basically not far removed from those described in literature” [Almefelt et al. 2006].

Some authors extend requirements management to incorporating the needs of variant specification. E.g. McKay et al. postulated early that already the requirements process needs to yield information on how to specify the variants needed in a market without generating data redundancies [McKay et al. 1996]. An implementation of such an approach is e.g. the model by [Harlou 2006]. More generally, it is often not sufficient to describe or model the requirements but also the variant drivers and their relationships to obtain a perspective on the product portfolio that is to be designed to obtain a better basis for the development of commonality and product architectures [Bühne et al. 2004].

4. Description of the integrated product information model

As a core concept, a “Product Information Model” was designed based on the requirements stated in Table 1. The model should serve as a basis for process re-design, i.e. to make all process descriptions consistent and define the common data elements to all processes. Based on that, the product data management (PDM) system to support the engineering design process was to be implemented.

The primary objective was to create a logic to best integrate all relevant data and their structures for the improvement of the related processes. This resulted in the product information model shown in Figure 2. It interrelates all structures, clarifies responsibilities, and unambiguously defines the interfaces between the involved disciplines (remark: In order to improve readability, specific terms of the information model are in the following written in italic).

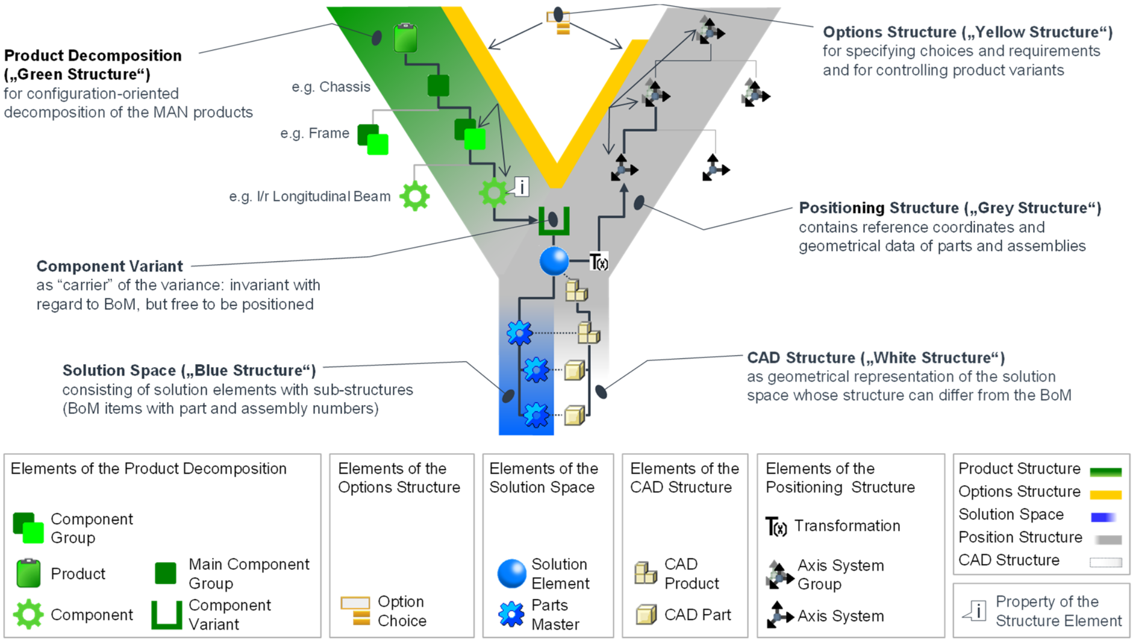


Figure 2. Structures and elements of the product information model

The Information Model clusters and interrelates structures used in various systems domains by different disciplines contributing to the product creation process in the early design phases:

- The *Options Structure* (“*Yellow Structure*”) contains product properties whose characteristics can be chosen by the customer. These characteristics are further specified through associated requirements. Released *Choices* are adopted as *Sales Codes* by the sales and distribution system. Planned *Choices* are merged as spec codes into a “specification book” handed over to a development project.

- The *Product Decomposition* ("*Green Structure*") consistently structures the development scope for all products and projects in generic form, i.e. independent from actual part numbers. The decomposition follows a strict hierarchy in terms of *Main Component Groups* (e.g. Chassis), *Component Groups* (e.g. Frame), and *Components* (e.g. Longitudinal Beam).
- *Component Variants* are the central elements of the *Product Decomposition* structure. Only at this level the part variance is instantiated and managed. *Component Variants* are created in development projects. After their release they become the building blocks for the configuration of ordered vehicles. *Component Variants* are invariant with respect to the bill of material (BoM), but variable regarding their position(s) in the vehicle.
- The *Solution Space* ("*Blue Structure*") consists of *Solution Elements*. Each *Solution Element* including its sub-structures (BoM items and their part numbers) represents a concrete technical solution for a required *Component Variant* [Ziethen 2006].
- The *Positioning Structure* („*Grey Structure*") comprises the available coordinate systems (nested), the selection rules of the *Axis System* as well as the related *Transformations* and the positioning information for the *Solution Elements*. All this provides the basis for the later positioning of the *Component Variants* in the vehicle [Ziethen and Koehldorfer 2004].

As the concept was initially designed by a small team of specialists from engineering and IT and then gradually discussed in the company, the colour coding of the various structures significantly helped to make the concept better understandable to other stakeholders.

5. Variant design using the product information model

The following description shows how the information model and the process are interrelated and what data relates to the different requirements as listed in Table 1.

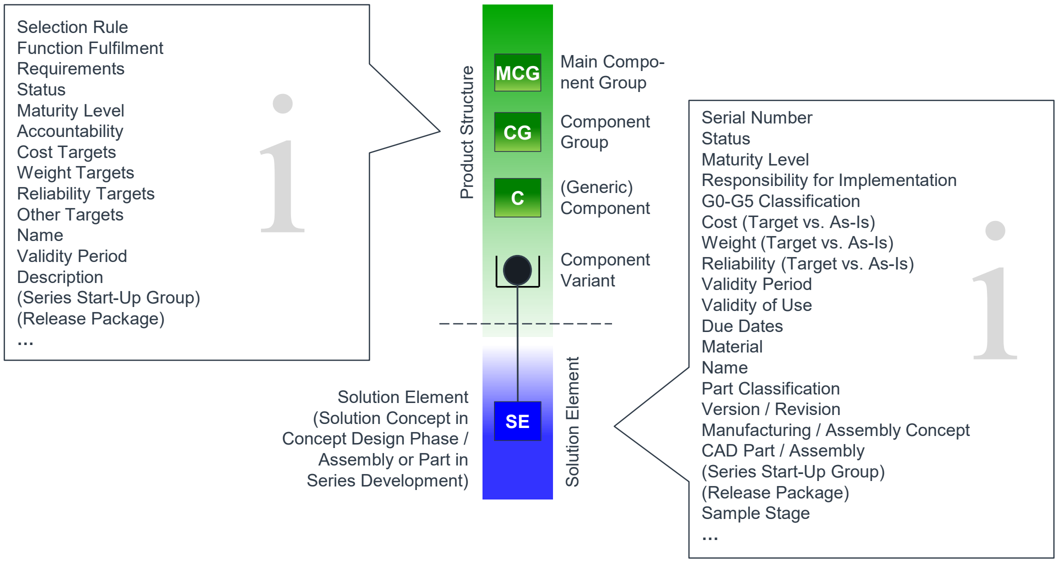


Figure 3. Product decomposition and solution space with assigned attributes (example)

5.1 Gradual "Growth" of data throughout the design process

During the "spec book" phase all involved disciplines state their requirements. These are either directly allocated to elements of the *Product Decomposition* or become assigned to spec codes. In both cases the same methodology is used for creating the links between the elements. It allows to navigate through the emerging structures, ensures traceability and helps to determine the degree of requirements fulfilment throughout the development process (state of "Pflichtenheft" implementation).

During the product creation process further attributes like status, maturity level, weight, part classification and many more are assigned to the various structure elements (see Figure 3). Due to this "ongoing (re)writing" of attributes in product development and subsequent phases, the generic

Components of the *Product Decomposition* and the *Solution Elements* (with part numbers) of the *Solution Space* become the data backbone of the product creation process.

5.2 Requirements description to support variant planning

The requirements are not simply described as a flat list of elements, but they are based on the same logic as the sales logic in the company. This helps to early model the intended functionality as required by the customer. Additionally, the requirements are structured in a functional manner, as e.g. all requirements to the “sun roof” are collected under a root node representing a *Choice*, e.g. “automatic sun roof”. Lastly, these *Choices* can be related to model their occurrence in the later product (e.g. “no sun roof with a spoiler”, or “seat heating always with leather seats”). Using Boolean operations, such data is the basic input for the later variant design. Figure 4 shows an example in the upper left corner: Leaf springs force a disc brake, and this relation then causes that no variant for a disc brake with leaf spring suspension is generated throughout the product design process.

Product managers thus talk the engineers through the “spec book”, which contains the *Choices* and the requirements. Each *Choice* is assigned to a “leading” *Component* for whose design one engineer is responsible. Along with this assignment the variant drivers for each *Component* are determined, documented and used to generate the *Component*-specific variant tree. Applying a first set of selection rules to these input data results in the target variance for each *Component*.

5.3 Product architecture design process

Based on the requirements associated to the *Product Decomposition* structure, the design of the *Components* and *Component Variants* is the next major steps. Figure 4 illustrates an example of how this takes place in the information model.

The figure shows how different *Component Variants* (in short: CV) are planned based on the requirements associated to the overall *Component*, in this case the second rear axle (“2nd RA”). The variant description at the requirements level forms a basic variant tree that is refined based on the technical solutions: *Component Variants* result from the linking of *Option Structure* elements to *Components* inside the *Product Decomposition* structure. Concrete *Choices* in the form of spec codes and requirements with target values initially lead to a high number of design cases (“fully expanded variance”). In a first step the application of a set of selection rules reduces the number of variants to those actually needed. In a second step technical and economic considerations further scale down the product variance to an “optimum”. The optimum is obtained by estimating take rates and by optimizing the degree of modularity for each *Component* in an iterative manner.

Once the *Component Variants* are planned, they are submitted to the actual design process and fulfilled with technical *Solution Elements* (i.e. part numbers). In the example shown in Figure 4 the *Solution Elements* are represented by the "balls" below the U-shaped *Component Variants* (Figure 4).

5.4 Variant coding

While the architecture process focuses on the required *Component Variants*, the variant coding generates a structured bill of material based on the required variance. Each confirmed *Component Variant* becomes the representation of a part of the overall bill of material. Any complete product is later configured by combining those *Component Variants* that – through their Boolean logic – can be combined without obstructing any of these rules. In order to enable this selection, each *Component Variant* carries an expression (again, in Boolean language) describing conditions, under which a particular *Component Variant* can be used. In addition to the requirements associated before, this expression carries the design rationale, i.e. the reasoning behind the variant and the usage it can be applied to. Typically, the expression will be derived directly from the architecture process and the *Component Variants* planning as shown in Figure 4.

Sometimes, the *Solution Element* fulfilling a planned *Component Variant* can be used for conditions that are currently not planned, but might be relevant in the future, e.g. a frame suitable for ten ton vehicles might also be suitable for twelve ton vehicles. In this case, the variant coding can already be extended with a view to future *Choices*. The resulting expression of the related *Component Variant* becomes the "carrier" of this knowledge.

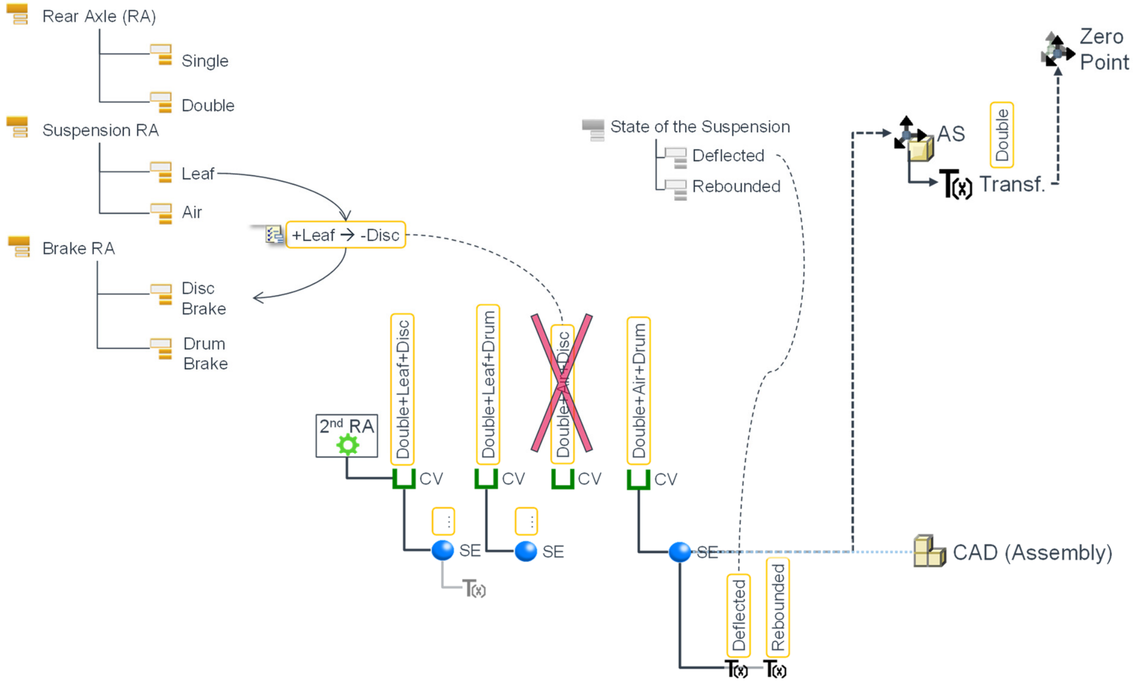


Figure 4. Deriving required component variants and coding logic

Of course, *Choices* can change over the time, e.g. from "fuel distance greater than 350 km" to "fuel volume 250 litres". Therefore, the *Choices* evolve in parallel through a managed process. Unreleased spec codes are discussed between product architects and design engineers, adapted if necessary and finally released to become sales codes in the sales and distribution system. Step by step the final number of *Component Variants* including the full set of selection rules is defined. In addition, the released spec codes and confirmed *Component Variants* together with the *Solution Elements* created for them provide a quite precise basis for tracking and reporting the progress of the development project in the concept design phase.

Later on, the released spec codes are transferred as "codes" to downstream IT systems including the sales configurator. Thus, out of the initial combinatorics defined in the spec book the final set of combinatorial rules is created while developing the product. These rules determine both from a market and from a technical perspective, which codes can/must/must not be combined.

5.5 CAD data management

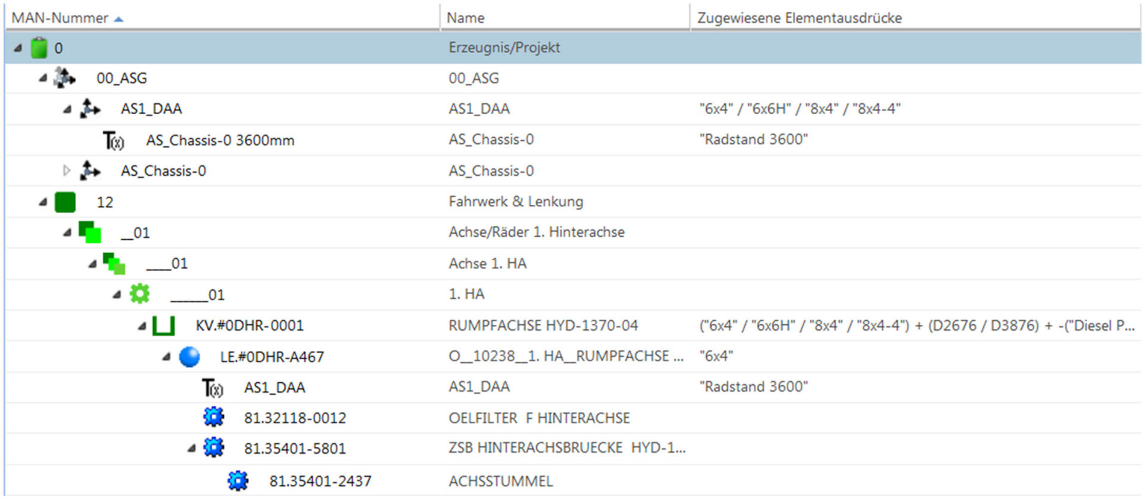
During the development process *Solution Elements* with part numbers are allocated to the *Component Variants*. Each of these *Solution Elements* represents the topmost assembly modelled in the CAD system for the corresponding *Component Variant*. In this manner, the *Component Variants* obtain their BoM elements, their geometry and their positioning information.

The positioning information consists of two parts: The first part dynamically determines for a selected set of *Choices* a parallel structure of coordinate systems including the *Transformations* to each other. This enables the engineers to think in installation spaces. This is shown in Figure 4 in the top right corner: The *Axis System* (AS) and its *Transformation* (T) are selected according to the particular *Choice*, e.g. "double" for a double (bogie) rear axle.

The second part provides the allocated *Solution Elements* with a selection rule regarding their positioning. This rule indicates, which coordinate system inside the installation space becomes the reference point and which *Transformation* positions the *Solution Element* inside the vehicle. In Figure 4, this is represented by the *Transformations* associated to the *Solution Element*. For instance, the structure of the coordinate systems differently positions the rear light depending on the type of the selected rear axle (single or double). Similarly, the distance of the rear lights can be adapted by referring to a different *Choice* (single or double) in the selection rule for positioning, which triggers a different

Transformation for the *Solution Element*. Additionally, these *Solution Elements* can carry alternate representations (which result in different CAD assemblies), e.g. two different states of the suspension ("deflected" or "rebounded").

Figure 5 shows how this data is managed in the company's PDM environment. The top rows represent the *Axis Systems* and *Transformations* in the left columns as well as the associated expressions and *Choices* in the right column. Underneath the *Product Decomposition* is resolved in terms of the *Main Component Group*, the *Component Groups*, the *Components* and the chosen *Component Variant* (the U-shaped element) with its expression on the right. Under the *Component Variant*, the fulfilling *Solution Element*, the associated *Transformation* and finally the parts are listed.



MAN-Nummer	Name	Zugewiesene Elementausdrücke
0	Erzeugnis/Projekt	
00_ASG	00_ASG	
AS1_DAA	AS1_DAA	"6x4" / "6x6H" / "8x4" / "8x4-4"
AS_Chassis-0 3600mm	AS_Chassis-0	"Radstand 3600"
AS_Chassis-0	AS_Chassis-0	
12	Fahrwerk & Lenkung	
__01	Achse/Räder 1. Hinterachse	
___01	Achse 1. HA	
_____01	1. HA	
KV.#0DHR-0001	RUMPFACHSE HYD-1370-04	("6x4" / "6x6H" / "8x4" / "8x4-4") + (D2676 / D3876) + -("Diesel P...
LE.#0DHR-A467	O__10238__1. HA__RUMPFACHSE ...	"6x4"
AS1_DAA	AS1_DAA	"Radstand 3600"
81.32118-0012	OELFILTER F HINTERACHSE	
81.35401-5801	ZSB HINTERACHSBRUECKE HYD-1...	
81.35401-2437	ACHSSTUMMEL	

Figure 5. Example of data objects in the PDM system

6. Validation results

The validation approach consisted of various steps:

- Early demonstrators using legacy data in spread sheets and animated presentations
- PDM prototype by means of a database application verifying and validating the product information model in smaller vehicle (component) projects [Kreimeyer 2012]
- Pilot study using both the PDM prototype and functionality of the final PDM solution over three years along with larger engineering design projects (ongoing): Approx. 1/3 of the entire MAN truck portfolio, 20,000 nodes in the *Product Decomposition*, 12,000 requirements
- Full usage of the final PDM solution based on the described product information model in a new vehicle design project with 10,000 new component variants, 1,000 choices to be newly designed, and 400 engineers involved

The overall validation proved that the approach described in this paper is functional and working in a productive environment. However, the validation results also showed that the complexity of the approach is rather high and that – while data can be modeled as required – the possibilities of "misuse" (i.e. modeling data not as desired) are very high. An intention, for example, was to early model reference vehicles for DMU checks using the final data structure. However, engineers use the data model to create choices like "reference vehicle 1" instead of the actual configuration data. As a follow-up need, those unwanted usages of the information model are now being closed off bit by bit.

A desired effect was to install an early variant planning process. Especially the separation of variants as "market variants" (modeled as *Choices*), "technical variants" (modeled as *Component Variants*), and positional variants (modeled through the reference *Axis Systems* and *Transformations*) provided a much better picture of when and where variants occur. Now, in a second step, more dedicated variant management processes are being installed together with sales, production and engineering to counteract an unmanaged growth of variants in each of these domains. Due to the implemented product information model, the variants can now be counted and reported. This functional possibility is key to a target-oriented variant management.

Thirdly, it is now possible to have a very early overview of the completeness of information during the early phases of design, e.g. the completeness of each data structure can be estimated and measured independently. The current vehicle design project is the first project that concludes the concept design phase with a clear picture of all variants that were in fact intended, designed, modeled and released.

An important organizational aspect was underestimated and it is now receiving the highest priority: The effort required for managing changes to the data. While the data model and its implementation bring more consistency to the overall design process, they also interlink the data much earlier in this process (see also "desired target state" in Figure 1). As the required information is often unclear or uncertain at this stage, it is often necessary to work with incomplete or preliminary data. This requires workarounds allowing the use of the information model without having access to the finally released data yet. Here, many problems occurred with impact on the overall organization of the design process. The earlier planning of variants necessitates new roles (who manages what changes?) and additional resources with a dedicated focus on the technical concepts, their modularity and their interfaces.

Finally, although the model was designed to be easily understandable through e.g. the color scheme as well as the dedicated role-based trainings and coaching that each user is put through, practice shows that users are generally somewhat overwhelmed, especially if users have a lower educational level (e.g. only CAD or drawing specialists with a vocational training background) or with older staff (e.g. if people have worked with “classic” 3D-CAD for the last decade or longer). Here, challenges lie mostly in the very integrative approach that forces people to think more outside their box, to earlier integrate other disciplines and thus to change their established work methodology and habits.

7. Conclusion and further work

7.1 Evaluation of the product information model from a scientific perspective

The chosen approach shows that an integrated product information model is possible in a large manufacturing firm. However, the complexity of the implementation is extensive. No such model can work without adequate IT support and process-focused implementation. These prerequisites are often underestimated, and in the scientific resources used to build the presented model, such aspects are either hardly or not regarded at all.

The validation phase confirmed the value propositions brought forth by the V-Model and by systems engineering (early top-down functional and technical planning). The chosen approach combines both with a variant-focused design process. Here, from a practical perspective, a gap in scientific approaches became very obvious: Literature either focuses on requirements-based engineering (e.g. as part of VDI 2206 or VDI 2221) or on variant planning and management (e.g. as [Erixon 1998] shows), but they rarely combine the two aspects. This gap was closed with the product information model.

The model can be generalized and therefore be applied to comparable companies. Expert interviews and the general exchange with other German manufacturers of agricultural and construction machines have shown that many such companies use similar approaches that, however, are hardly as comprehensive and consistent as the one shown here. Specifically, most companies use a similar setup of their product structures description, which confirms the conceptual core of the described model.

7.2 Scientific implications

The successful implementation also shows that a scientific approach to an industrial problem can be useful. The described product information model was developed with academic support [Kreimeyer 2015] over more than a year. This often caused criticism within the company. However, the theoretical concept helped during the implementation period to guide all detailed discussions and to provide the conceptual boundary conditions against which all details of the IT implementation were considered. Often enough, the limits of the abstract model were revealed during the discussion with practitioners. These discussions helped to enrich the abstract model and to enhance the overall concept. Nevertheless, having and using the model was vital to keep the discussions on track and not to loose the overall goal out of sight.

7.3 Next steps

As the presented product information model was implemented and applied to a first "real" project, next steps now consist in completing the model with regard to the needed change management processes. Once this is available, the PDM system supporting the model will be deployed throughout the company. As part of this deployment, the specific needs of each engineering division are compared with the available functionality to ensure that the demands of all engineering divisions are met.

References

Almefelt, L., Berglund, R., Nilsson, P., Malmqvist, J., "Requirements management in practice: findings from an empirical study in the automotive industry", Research in Engineering Design, Vol. 17, 2006, pp. 113-134.

Blees, C., "Eine Methode zur Entwicklung modularer Produktfamilien," Dissertation, Technische Universität Hamburg-Harburg, Hamburg, 2011.

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

Erixon, G., "Modular Function Deployment: A Method for Product Modularisation", Dissertation, Department of Manufacturing Systems, Assembly Systems Division, The Royal Institute of Technology, Stockholm, 1998.

Friedel, A., "Investigating the Management of Uncertainty in Product Platform Lifecycles", Master Thesis, Engineering Systems, MIT, Cambridge, MA, 2011.

Harlou, U., "Developing Product Families based on Architectures", Dissertation, Department of Mechanical Engineering, Technical University of Denmark, Copenhagen, 2006.

Jose, A., Tollenaere, M., "Modular and platform methods for product family design - literature analysis," Journal of Intelligent Manufacturing, Vol. 16, 2005, pp. 371–390.

Kipp, T., "Methodische Unterstützug der variantengerechten Produktgestaltung", Hamburg, TuTech, 2011.

Kreimeyer, M., "Implementing Product Architecture in Industry: Impact of Engineering Design Research", Impact of Design Research on Practice, Chakrabarti, A., Lindemann, U. (eds.), Springer Berlin, 2015, pp. 383-398.

Kreimeyer, M., "Innovatives Prozess-Prototyping", Quo vadis, Automobilindustrie 2020, 2012.

McKay, A., Erens, F. J., Bloor, M. S., "Relating Product Definition and Product Variety", Research in Engineering Design, Vol. 2, 1996, pp. 63-80.

Robertson, D., Ulrich, K., "Planning for product platforms", Sloan Management Review, Vol.39, 1998, pp. 19-31.

Schuh, G., Arnosch, J., Nußbaum, D., "Produktarchitekturen richtig gestalten: Ein Weg zum variantenoptimierten Produktprogramm", Industrie Management, Vol. 6, 2007.

Ulrich, K., "The role of product architecture in the manufacturing firm", Research Policy, Vol.24, 1995, pp. 419–441.

Ulrich, K., Eppinger, S., "Product design and development", McGraw-Hill/Irwin New York, 2012.

Ziethen, D., "CATIA V5 Baugruppen, Zeichnungen", Hanser München, 2006.

Ziethen, D., Köhldorfer, W., "Konstruktionsmethodik zur Modellierung von Volumenkörpern", Hanser München, 2004.

Dr. Ing. Matthias Kreimeyer, Centralien / Senior Manager Product Architecture
MAN Truck & Bus AG, Architecture
Dachauer Str. 667, 80995 Munich, Germany
Email: matthias.kreimeyer@man.eu