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

MANAGEMENT OF VEHICLE ARCHITECTURE PARAMETERS

F. Toepfer and T. Naumann

Keywords: parameter management, vehicle architecture, systems engineering (SE)

1. Introduction

Product development in the automotive industry is increasingly driven by global competition to acquire demanding customers and stringent C02 requirements resulting in product variance, individualization, function-integration and novel technologies. To fulfil these diverse requirements under the increasing pressure of time and cost, vehicle architectures are used to systematically realize module- and common-part-strategies.
Aside from economic advantages, these strategies result in an immense increase of technical, organizational and procedural dependencies in a highly complex development environment. In order to cope with this high-level complexity in product development, Systems Engineering-related model- and modelling-approaches are pursued. They systematically deal with dependencies in order to achieve better transparency and to avoid inconsistencies along the development process. These approaches however lack of broad acceptance among designers mainly due to high modelling effort and the unwillingness of designers to learn and use additional and particularly complicated (modelling) tools. This paper therefore addresses the research question, how complexity in product development can be handled with appropriate tool-support in a way that it provides a process-supporting benefit for designers instead of causing additional work.
The approach described in this paper, facilitates the process-spanning management of functional and geometric parameters in a database-application. This data base approach allows a qualitative management of parameters interdependencies by defining of so named active chains. Furthermore, the database-application can be coupled to CAD-models allowing for an initial integration of CAD-Parameters and their continuous update in the database.
The management-approach depicted hereafter marks a novel, additional approach in Systems Engineering. It is enabling bottom-up modelling of a system model as well as process-spanning traceability of parameters. To validate this Systems Engineering-approach a first application was conducted and tested in the context of vehicle architectures. Therefore, the significance and use of vehicle architectures is described in chapter three following an introduction of terms and definitions. Current approaches in Systems Engineering aiming for the management of complexity are outlined in chapter four including challenges these approaches are facing and thereby highlighting the necessity of a novel approach as presented in chapter five.

2. Terms and definitions

To understand vehicle architecture and in order to elucidate interdependencies in vehicle development, the following section addresses basic terms and definitions.

Complexity

Complexity generally is referred to as a system property characterized by the number and variety of elements and their relations as well as the multitude of possible states [Ulrich and Proust 1995], [Ehrlenspiel 2013].
Product development itself can be seen as a sociotechnical system that incorporates different aspects of complexity [Ropohl 2009], [Naumann 2014]. From a technical perspective complexity is driven by the number of parts and functions in the product itself and their interactions. The development of a technical system is executed by a great number of individuals with different responsibilities and their continuous interaction to finally assemble single parts to one product that realizes the required functions. With an increasing level of technical complexity the organizational complexity rises simultaneously [Naumann 2005]. In the development organization a multitude of disciplines are working together all bound to their own and to shared processes. Finally procedural complexity is increasingly driven by a high parallelization of processes due to a reduction in development time.

General definition of a product-architecture

A technical product realizes a function. In order to do so, the product is characterized by a physical setup consisting of elements which transfer inputs into outputs with a potential change in state. A product thus can be described on both a functional and a physical level.
By decomposing a product's overall function, a function-structure is generated allowing a close-up examination of sub-functions. In the same way the physical layout of a product can be described by a product-structure [Feldhusen and Grote 2013].
An architecture describes how a product's function is realized by its physical elements. Designing an architecture thus means assigning functional elements to physical elements and defining their mutual interfaces [Ulrich 1995].

Vehicle architecture

A vehicle architecture respectively describes the physical layout of a vehicle and the way it realizes its function by a given set of basic architectures parameters and modules. In practise the term vehicle architecture is synonymously used for a product- or product series-spanning modular architecture.
The significance of a vehicle architecture arises from the fact that every vehicle has an architecture, but that an architecture is not limited to one particular vehicle. By purposefully developing a vehicle-architecture, it can be used within different products a product series. It thereby advances to a key factor in effectively and profitably developing product-variants. Thus, every customized vehicle ideally comprises a standardized, product-series-spanning architecture, a product-specific platform and product-variant specific, individual characteristics.
The aim behind developing a product-spanning architecture is to be economical. A modular layout incorporates standardized interfaces which allow common-part strategies throughout different product-variants and product-series leading to economies of scale. By reducing module variants, costs in development, purchase and production can be lowered as well.
A modular architecture-layout allows an architecture-spanning reuse of existing modules, the exchange of newly developed modules, the extension by adding modules as well as creating variants by a combining module. Modules which are relevant for the vehicle architecture are at least identical in concept and underlay a communality strategy within a platform.

Modularization, modularity and modules

Modularization is used to simplify a complex system by reducing the number of system elements and their relations respectively their interdependencies among one another [Göpfert and Steinbrecher 2000]. Modularity thus determines the degree of independence in between a systems' elements [Ulrich 1992]. Functional independence thereby describes to what extent functions can be realized independently from other modules. Physical independence enables the physical separation of modules by purposeful designed interfaces [Göpfert and Steinbrecher 2000].

A module is therefore a component or an assembly characterized by weak interdependencies to other components or assemblies. Modules have well defined and mostly standardized interfaces and can differ in their function. Defined interfaces facilitate an easy exchange of modules and allow for an easy creation of variants by module combination [Schuh 2012].

Product series, platform, modular system

Product series

A product series describes a multitude of technical systems differing in scale but realizing functions based on the same solution principle [Schuh 2012].

Platform

Platforms are used to effectively create variants. A platform is a neutral basis for all variants. Consisting of a defined core of standardized elements, variants are created by attaching individual elements [Steffen and Gausemeier 2007], [Feldhusen and Grote 2013].

Modular system

The combination of functional elements within a modular system allows for the building of function-variants. A modular system thus describes an entity of assemblies, single-parts or modules which allow for the creation of variants with differing overall functions within a free combination system. [Steffen and Gausemeier 2007].

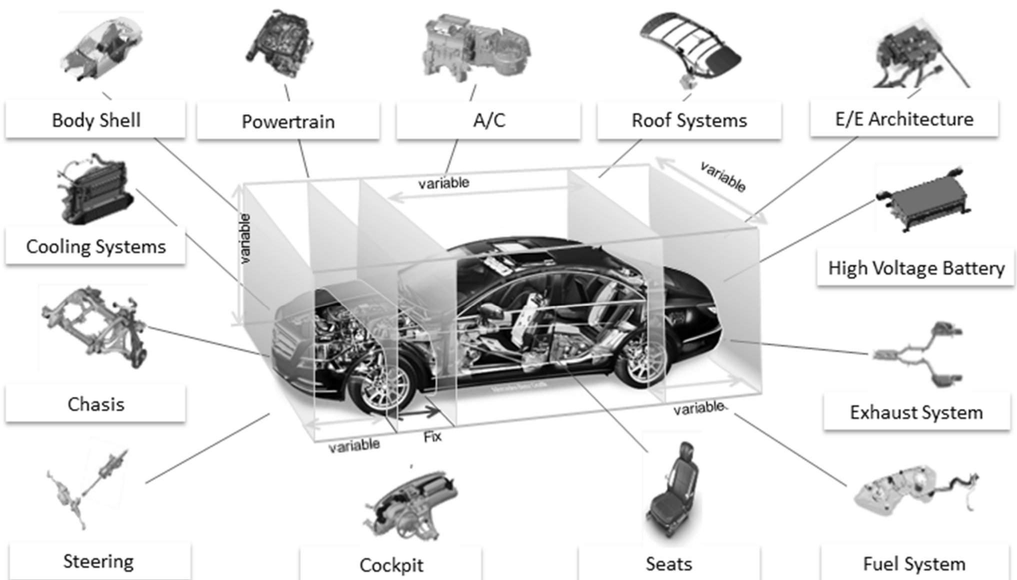

Figure 1. Vehicle architecture with common modules

3. Development of a vehicle architecture

In order to provide process supporting tools, processes in the development department of Daimler AG were analysed. In addition, interviews with designers were conducted which outlined difficulties in current processes and revealed sources for data inconsistencies.

The architecture process describes the development of an architecture which takes place before and during the process of variants development. In order to define architecture suitable for all later variants, requirements of variants have to be anticipated and considered in the architecture layout process. [see also Schuh 2012].

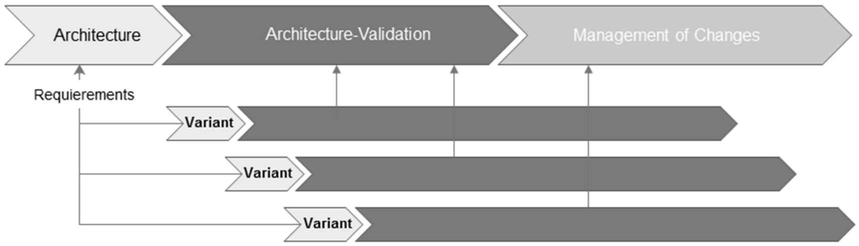

Figure 2. Development process of an architecture

Developing and validating an architecture is realized by the cooperation of many disciplines. Figure 3 shows relevant aspects in the validation of the architecture leading to multitude of goal conflicts.
Validations take place among many disciplines within the entire development organisation. Every discipline has different requests for a solution which can compete with those of other disciplines. Finding a solution suitable for all disciplines requires their willingness to find a reasonable compromise.
The core challenge in using a product series spanning architecture is the increase of procedural complexity. Modules and components which are integrated in different product variants have to be validated in additional contexts. These additional boundary conditions lead to an increase in product-specific interdependencies. Bringing the development of different product series together, leads to increasing interaction between more developers. At the same time, organisational and procedural interdependencies are driven forward by strict quality requirements to minimize the financial risks in case of deficiently developed components. A systematic handling of interdependencies is therefore essential in product development.

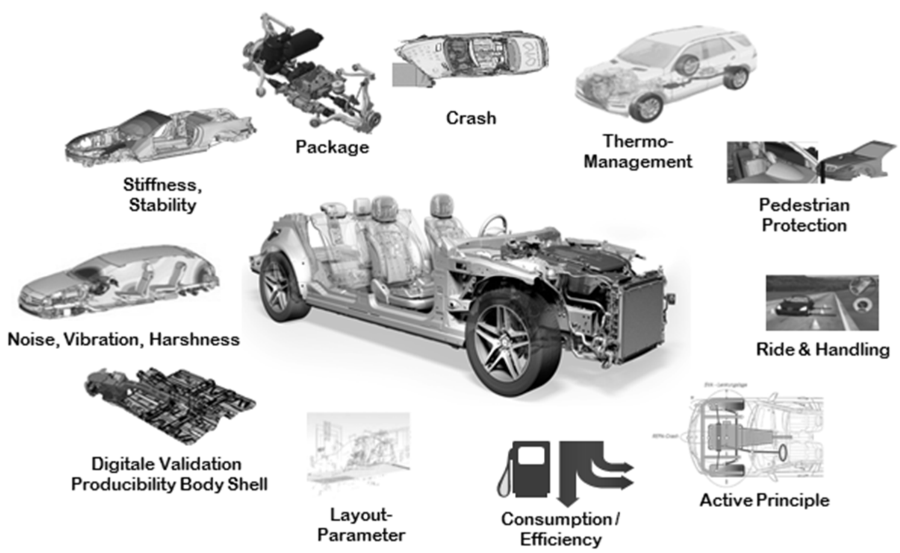

Figure 3. Relevant aspects in architecture-validation

4. Current approaches in systems engineering

As outlined before an utterly complex system like product development consists of an unmanageable amount of interdependencies. Knowledge about these interdependencies leads to better transparency of changes and allows for a better estimation of impacts that come with changes [Koenigs et al. 2012]. According to INCOSE, Systems Engineering is an "[…] interdisciplinary approach to enable the realization of successful systems." To effectively handle complexity in product development, system models are used to trace interdependencies within a highly dynamic system. The extension of Systems Engineering by a centrally available, digital system model is therefore referred to as Model based Systems

Engineering and is realized by the use of modelling languages with a stringent semantics and syntax to distinctively describe the model.

Requirements for system modelling

System modelling allows for the representation of systemic interdependencies within a system model. In the context of Systems Engineering, traceability marks one of the central aspects. Traceability describes the ability to describe, document and trace interdependencies among artefacts along the entire development process. According to the six dimensions formulated by Ramesh and Jarke [2001] a modelled dependency in between two artefacts should give information about:

- What type of information is recorded?
- Who created or changed the artefact?
- Where the information is from?
- How is the information depicted?
- When was the artefact created or changed?, and
- Why was the artefact created or changed?

Managing this information offers a wide range of advantages for product developers participating in the development process such as easy identification and verification of impacts due to changes, improvement in the communication of changes, documentation of changes and the reuse of knowledge [Koehler et al. 2014].

Challenges in systems engineering

Nowadays a multitude of prototypical and a handful of industrial tools exist, which allow for system modelling and traceability. Koenigs et al. [2012] identified a vast modelling effort as one of the major deficits in the acceptance of traceability in practice. Furthermore Storga [2004] describes difficulties originating from the variety of traceability needs in different projects, organizations and users due to different goals in specific tasks, complexity of the knowledge base which leads to unmanageable effort in creation and maintenance of full traceability, the integration of both formal and informal sources that is necessary and the limitation for traceability due to existing heterogeneous tools and methods.
Further criteria for suitability, selection and application are:

- Representable system complexity,
- Low effort in modelling (compared to procedural and organizational benefit),
- Easy adaption of information in the model,
- Instantiation of the models,
- Integration into the existing processes and the IT-environment.

The degree to which system complexity can be represented in a model relies on the type of data and the type of relation that is to be coupled. Information about interdependencies in a model can be limited to a pure qualitative character or at the same time contain quantitative character.
High modelling effort and an elaborate adaption of models are essential factors in a generally low acceptance of system modelling approaches. Disciplines dealing with mechanical construction in particular supress functional aspects due to geometrical layout. However, geometrical interdependencies are already part of the existing CAD models and product configurations. This fact increases the difficulty to initially make functional aspects explicit at all. In addition to that, the user is questioning the efficiency of redundantly modelling interdependencies already existing in the CAD- and PDM-system environment.
Acceptance, suitability and application of system modelling are additionally influenced by the assignment of roles and responsibilities. Benefits from a holistic system model can be gained in particular for comprehensive aspects and responsibilities. In automotive development these responsibilities are represented by the departments of advance development and full vehicle development.
The department of advance development is responsible for the basic functional layout as well as for the definition of the architecture. The division of full vehicle layout is then responsible for the functional and geometrical integration and validation from a concept design until maturity. Hence, both divisions

operate in the context of the overall architecture respectively the overall vehicle. Opposing to the profiting departments, detailed information about functional and geometrical interdependencies have to be gathered and displayed by developers responsible for single components or modules. They most often don't profit from the holistic system model. Knowing the relations and interdependencies in between their own components, system models haven't had an influence on their work so far.
The failure of current modelling approaches is mostly due to a disproportional modelling effort. A novel approach has to avoid this failure by offering a low-effort and easy to use management functionality. Furthermore, methods have to be developed which avoid the redundancy among different models.

5. Approach of parameter management

Following the analysis of current processes and taking into consideration the challenges of current Systems Engineering approaches a prototype for a process supporting Systems Engineering Tool was developed to systematically support processes in product development.
The presented approach is based on a database solution which allows the management of parameters relevant for the development process. Their interdependencies can easily be modelled in form of active chains. A parameter generally describes a characteristic variable of a system element or an element relation which can possess multiple instances in different product variants. These parameters can be of functional or geometrical character. The approach allows for the definition of parameters at any time within the tool itself. However, geometrical parameters are already implemented in CAD-models. These models again are managed in product-specific configurations in product data management-systems (PDM).

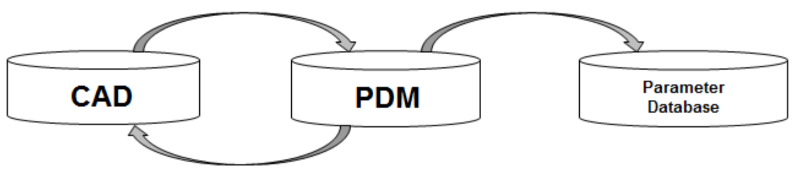

Figure 4. Integration of a parameter database into the CAD-PDM-Environment

The application for the novel parameter management approach is therefore coupled to the PDM system and thus facilitates a direct link to parameters implemented in the CAD-models. CAD and PDM are integral parts of the existing IT-environment. By coupling the CAD- and PDM-system, parameters are not only managed without redundancy but also kept up to date. Changes are thereby kept transparent. Since every parameter is part of a CAD-model, thereby part of a PDM-system and once again part of one or multiple product configurations, the following information automatically become content of the system model:

- Name, ID and shape of the system component,
- Creator and editor,
- Context of application (Product Series, Module, Assembly),
- (Parametric) geometry,
- Editing, Release and Changes,
- Variants and Versions.

In order to retrieve and interpret parameters in CAD data, a machine-readable information of the parameter and its value has to be embedded. Using XML-technology is one way to implement this functionality.

Keeping CAD-Data consistent

For the consistent interpretation of position-dependent parameters, transformation-matrices have to be considered. Depending on its reference coordinate-system a random point in design space can be described by different coordinates. A point described by a components coordinate-system, possesses different coordinates when described by the global vehicle coordinate system as long as both coordinate-systems aren't identical. To consistently depict positional parameters differing coordinate-systems and their relations have to be considered. Figure 5 schematically describes the transformation from component- and assembly-coordinate-systems to the vehicle coordinate system.

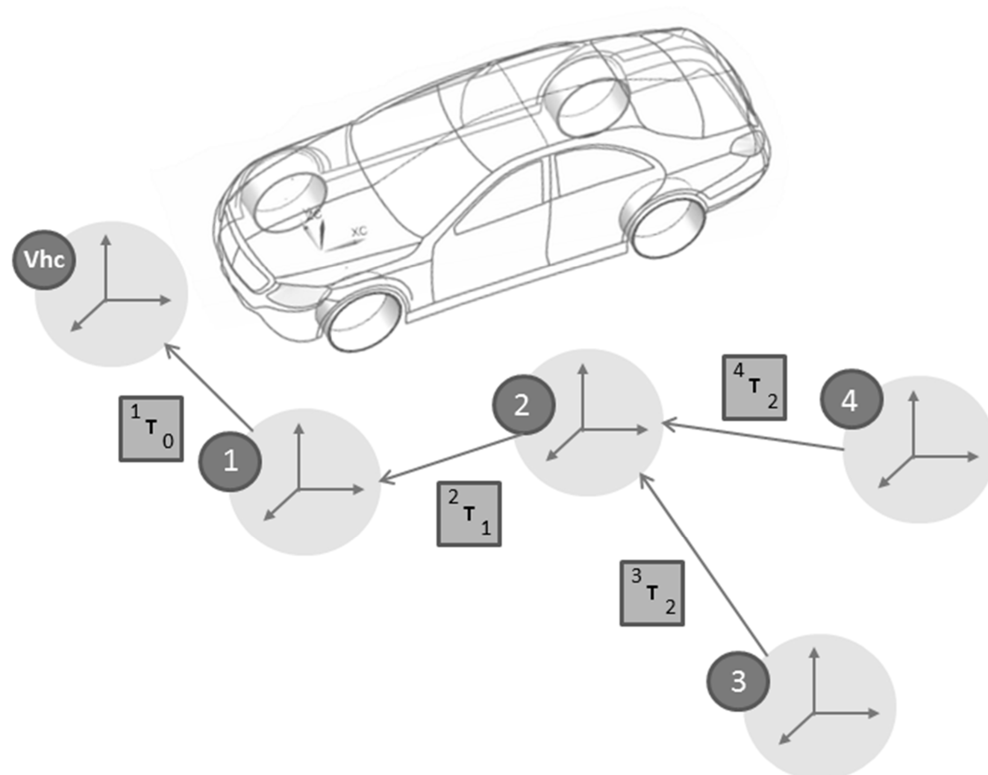

Figure 5. Transformation of local coordinates into global coordinates

Against the background of a product-series-spanning architecture, requirements of the architecture which have to be maintained in all product variants can be documented and traced. In case of deviations from the requirements, the application outlines the inconsistency. The parameter database is not limited to the documentation architecture requirements which have to be kept consistent but can be used for any kind of relevant parameters in CAD-models and their model-internal and model-spanning interdependencies, such as concept parameters, parameters of specific modules or parameters used in task-specific templates.

Active chains

Functional and geometric interdependencies within and in between components can be documented by the coupling of component-specific parameters within an active chain. An active chain ought to be seen as a container where parameters are brought together. A parameter can be part of several active chains as well. Once a parameter changes it may affect any other parameter that is part of the active chain or even parameters in other active chains if the particular parameter is related to more than one active chain. Figure 6 shows schematic interdependencies of component specific parameters, comprised in either one or multiple active chains.

The presented approach is neither limited to a specific process nor to a systems' level of complexity. Parameters can be defined at any time in the development process and linked to any other parameter.

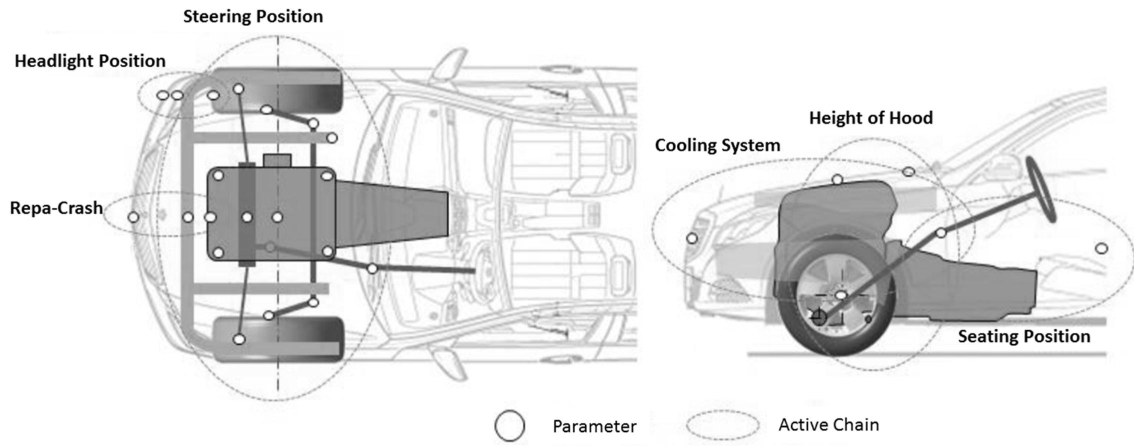

Figure 6. Assignment of component-specific parameters to active chains

6. Application

To verify the functionality of the described approach, different use cases with concept-, template-, module- and architecture parameters were implemented of which the latter is outlined in this paper. However, interrelations in between these different types of CAD related parameter are given and can thus be connected in active chains. Figure 7 shows a simplified CAD model of an architecture designed for the purpose of validation.

Parameters (in CAD-models referred to as Expressions) relevant for the architecture were made explicit within the CAD-model and subsequently archived into the PDM system. All parameters were then extracted from the archived models and transferred to the database. The assignment of parameters to predefined products, components and active chains was realized in the application.

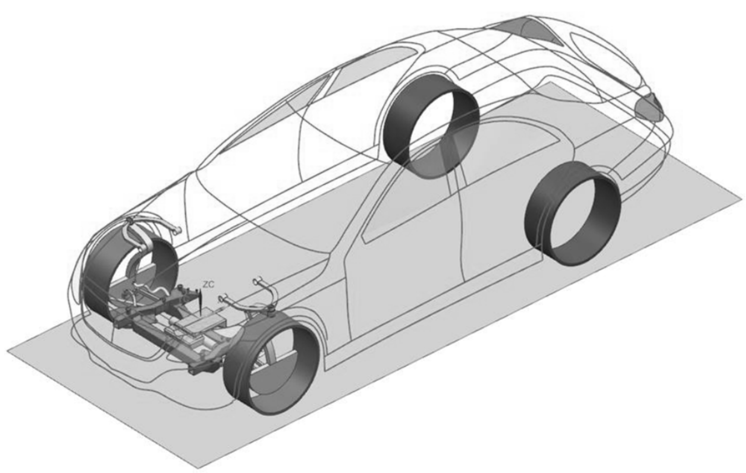

Figure 7. Test model for a vehicle architecture

Developing a vehicle architecture means designing a functional and geometrical full vehicle layout under the consideration of communality and product specific individual characteristics. From a todays point of view, the development of architectures is characterized by the takeover of existing concepts.

With the strategic specification of architecture-relevant modules and communalities the integration into existing product structures takes place. After that an iterative validation of the specific concept takes place for every product variant. Once the full vehicle layout is not applicable to a product variant the architecture has to be adopted for all product series.

The parameter management facilitates a transparent communication and traceability of changes in product variants. It is thereby not limited to parameters in the full vehicle context. It can be used for a multitude of applications with different levels of detail.
Concept parameters for example can be implemented into the database at a very early phase of the design process. These parameters can then be traced over the entire process and offer input for further templates or construction requirements. Through the coupling of parameters in active chains architecture parameters originating from an architecture template can be coupled with any kind of more detailed CAD-model used to validate the architecture. A broad set of technical, procedural and organizational interdependencies is the result, offering comprehensive transparency.
The prototype of the database application is already being used in industrial practise for the purpose of a consistent and transparent architecture process. The feedback from designers directly involved in the architecture process is utterly positive as the tool offers up-to-date design information which can be traced from conception to the actual product series development. Even designers, responsible for the development of components and thus only providing information from their CAD-Data rather than directly profiting from the arising holistic system model, regard the approach in a favourable light as it causes only very small effort.

7. Outline

From a today's point of view, functional aspects remain mostly unconsidered in the early full vehicle layout. The adaption of the geometrical overall concept takes place within the implicit knowledge of functional interdependencies. The adjusting lever for functional manipulation however is too small due to an adopted and predefined product structure. A methodical development of complex systems is meant to be executed starting with a rough layout leading to a detailed concept. This process is ought to be supported by the use of models.
By using an architecture template the development of the architecture can be extended by functional aspects. An architecture comprises both, a product structure and a function structure. The function is carried out by physical components and their interdependencies.
Due to these interdependencies a change in product structure leads to a change in function. The consideration of functional changes is indispensable for a functional layout. Benefits from an architecture template result from the consideration of basic functional aspects in the early phase of the full vehicle layout.
To evaluate the impact of geometrical changes the geometrical model has to be coupled with a mathematical model. Functional quantities such as inertia or the centre of gravity can be computed. At the same time basic geometric validations can take place such as ergonomic aspects or packaging.
The depicted test model gives an impression of a possible layout of an architecture template. The template has to represent the essential modules of a vehicle. An easy adaption of the modules has to be considered in the construction and is essential to depict modules of a product series-spanning architecture. These modules are as mentioned before identical in concept but different in scale and can thus be adapted by the change of parameters. Furthermore, the layout should consider the easy positioning of components. An imaginable approach to implement this functionality is the construction of geometrical solids referring to a predefined offset coordinate system which allows a parametric positioning of components.
Aside from the computation of functional properties in the early architecture concept an optimization can take place as well. The overall function of a vehicle is defined by a multitude of sub-functions. These sub functions can be described in mathematical models. Changes in the value of parameters which are part of the mathematical model lead to changes in specific vehicle properties. Parameters ought to be understood as alterable quantities. Implemented in functions with defined boundary-conditions they span a defined set of solution spaces. Goal of the optimization is to define parameters in such way that the overall solution is part of an optimal solution space defined by the intersection of competing design solutions. With an appropriate interface the result of the optimization can be returned into the database.
The use of a parameter database for the management of relevant parameters marks a promising approach in creating transparency and consistency in a highly cross-linked development process. In particular, the crosslinking of parameters in active chains seems to be an appropriate low effort modelling approach

which will gain the acceptance of product developers. The potential for the management of functional parameters for an early consideration and optimization of overall vehicle functions is immense.

References

Eigner, M., Roubanov, D., Zafirov, R., "Modellbasierte virtuelle Produktentwicklung", Springer-Verlag, Berlin, Heidelberg, 2014.

Feldhusen, J., Grote, K.-H., "Pahl/Beitz Konstruktionslehre", Springer-Verlag Berlin Heidelberg, 8. Vollständig überarbeitete Auflage, 2013.

Göpfert, J., Steinbrecher, M., "Modulare Produktentwicklung leistet mehr", Harvard Business Manager, Heft 3, 2000.

INCOSE, "International Council on Systems Engineering", Available: <http://www.incose.org>.

Königs, S. F., Beier, G., Figge, A., Stark, R., "Traceability in Systems Engineering - Review of Industrial Practices, State-of-the-art Technologies and New Research Solutions", Advanced Engineering Informatics 26, 2012.

Koehler, N., Naumann, T., Vajna, S., "Supporting the Modelling of Traceability Information", International Design Conference, Dubrovnik, 2014.

Naumann, T., "Adaptives Systemmanagement", Dissertation, Otto-von-Guericke Universität, Magdeburg, 2005.

Naumann, T., Koehler, N., "Meta-Model of Sociotechnical Systems: Derivation, Structure and Content", Proceedings of the tenth international symposium in tools and methods of competitive engineering (TMCE2014), Budapest / Hungary, 2014.

Ramesh, B., Jarke, M., "Towards Reference Models For Requirements Tracebility", 2001.

Ropohl, G., "Allgemeine Technologie", Universitätsverlag Karlsruhe, 2009.

Schuh, G., "Innovationsmanagement", Springer-Verlag, Berlin, Heidelberg, 2. vollständig neu bearbeitete Auflage, 2012.

Steffen, D., Gausemeier, J., "Modularisierung mechatronischer Systeme", erschienen in Industrie Management 23, 2007.

Storga, M., "Traceability in Product Development", Proceedings of the 8th International Design Conference, Dubrovnik, 2004.

Ulrich, K., "Fundamentals of Product Modularity", Management of Design - Engineering and Management Perspectives, Springer Science+Business Media, New York, 1994.

Ulrich, K., "The Role of Product Architecture in the Manufacturing Firm", Research Policy 24, 1995.

Ulrich, H., Probst, G. J. B., "Anleitung zum ganzheitlichen Denken und Handeln: Ein Brevier für Führungskräfte", Bern, 1995.

Ferdinand Toepfer, M.Sc.
Daimler AG, Engineering IT
Hanns-Klemm Str.5, 71034 Böblingen, Germany
Email: ferdinand.toepfer@daimler.com