NordDesign 2022
August 16-19, 2022
Kgs. Lyngby, Denmark

Towards a Systems Engineering Methodology for Architecture Development of Vehicle Concepts

Jonas Krog[1], Tarık Şahin[2], Thomas Vietor[2]

[1]Vehicle Concept Development, Volkswagen AG
jonas.krog@volkswagen.de
[2]Instiute for Engineering Design, Technical University Braunschweig
tarik.sahin@tu-braunschweig.de; t.vietor@tu-braunschweig.de

Abstract

The automotive industry is undergoing a huge change, as the global trends digitalization, sustainability and urbanization are further accelerating. Cars are evolving from mechanical dominated systems to complex interdisciplinary systems. New market demands and complex functionalities include all engineering disciplines throughout multiple subsystems of the vehicle. Current industry practice shows that these challenges of complex and interdisciplinary development cannot be met yet. Existing processes and mindsets are often narrowed on single engineering disciplines or systems. Especially during concept phase, which has a big leverage on the further development, the current focus is mechanical, geometrical design. The theory of product development and Systems Engineering address these challenges and provide suitable approaches with structured product models. This contribution elaborates state of the art and adapts it to a new methodology to introduced Systems Engineering to vehicle concept development. The requirements for such methodology are derived, addressing the architecture development of vehicles in terms of Systems Engineering to handle interdisciplinary complexity, enable better consistency and traceability. Based on state of the art research, a product model with four views (requirements, functional, logical and physical; RFLP) is presented. It introduces the Vehicle Systems Architecture, which represents the new understanding of architecture development during vehicle concept phase and contains the important architecture perspectives. The interpretation and role of each view regarding automotive development is elaborated, including its application in the left branch of the V-model with a hierarchical decomposition in system levels. The presented idea of the methodology for vehicle architecture is to enable a consistent, customer- and system-oriented development of complex vehicles by implementing new architecture views.

Keywords: Systems Engineering; MBSE; automotive development; V-Model; product development; RFLP; vehicle architecture; system architecture; concept development

1 Introduction

The automotive industry is undergoing a huge change, as the global trends digitalization, sustainability and urbanization are further accelerating (McKinsey & Company, 2019; Schulze et al., 2020). Customer demands are becoming more ambitious and dynamic. Thereby cars are becoming more complex, especially more interdisciplinary, evolving from mechanical dominated systems to cyber-physical systems, interacting with their (digital) environment, fully equipped with sensors and actuators (Verein Deutscher Ingenieure, 2020). Most innovations and new functionalities within vehicles are implemented multidisciplinary with software, electric/electronic (E/E) and mechanics (Anderl et al., 2012). A multitude of systems work together to enable complex functionalities, like automated driving which involves several systems at once. This ultimately leads to a more challenging, more complex and interdisciplinary development process (Dumitrescu et al., 2021). Current industry practice shows that these challenges cannot be met yet, as the existing processes and mindsets are often narrowed on a single engineering discipline or system (Volkswagen AG, 2022). In many cases it is not possible to comprehend which functions and systems are addressing which customer requirement. Methods of Systems Engineering (SE) are needed, that facilitate interdisciplinary system development, allow a better customer- and function-oriented development and handle requirements more consistently to keep up with the dynamic markets. These methods are especially important for the early development phase, the concept phase, which has the biggest leverage to prevent issues and handle challenges throughout the further development process (Michels, 2016). As of today the vehicle concept development is mainly focused on mechanical, geometrical vehicle design, hardly considering interdisciplinary functionalities and systems. This publication elaborates existing methods from product development theory and Systems Engineering that address this purpose and suggests a methodology for improved vehicle architecture concept development.

2 State of the Art

The presented state of the art is based on literature research and observations in an industrial environment in the complete vehicle concept department at one of the largest automotive manufactures. The practical observations and conclusions are derived accordingly, as a shift towards SE practices is currently taking place (Volkswagen AG, 2022).

2.1 Vehicle concept development

The automotive development process consists of three main phases: product definition, series development and pre-production phase (Gusig & Kruse, 2010; Pischinger & Seiffert, 2021). Vehicle concept development is located in product definition to set the base and investigate the technical feasibility of a new vehicle. This phase plays an important role as changes, problems and complexity can be handled and avoided with relatively little effort (Feldhusen & Grote, 2013; Michels, 2016).

Today the ‘classic’ concept development focuses on the geometrical and mechanical design, based on a variety of first requirements (Gusig & Kruse, 2010; Pischinger & Seiffert, 2021). A rough geometrical concept is designed, driven by the vehicle design, vehicle parameters, main modules and components (Pischinger & Seiffert, 2021). The concept phase is finished with 3D virtual product models, defining the geometrical architecture and dimensions of the vehicle (Busche, 2014; Felgenhauer, 2019). This geometrical architecture ‘describes the physical layout of a vehicle […] by a given set of basic architectures parameters and modules’ (Toepfer & Naumann, 2016). The handover of representing CAD models to series development is relatively simple, on the contrary it is challenging to remain transparent with complex dependencies and consequential decisions during handover (Busche, 2014). Current vehicle concept development is limited on geometrical and mechanical architecture. Necessary functional and system perspectives are hardly used during concept phase, which is no longer sufficient considering the rising complexity from multidisciplinary systems and dependencies.

2.2 Theory of product development

To face the challenges of multidisciplinary systems research in engineering design has been advancing process models and theories. An established process model for the development of mechatronic or cyber-physical products is the V-model (Eigner et al., 2014; Hick et al., 2019). It describes the 'inherent factual [...] dependencies of the development tasks' (Verein Deutscher Ingenieure, 2020), covering the entire development process at an abstract level. Vehicle concept development can be located in the left branch of the V, which aims for the development of the overall system. The VDI 2221 norm describes the left V-branch more detailed, from function structures to physical solutions, which corresponds to the definition of product architecture in the theory of product development (Verein Deutscher Ingenieure, 2019). Here product architecture is defined as the combination of function structure and physical product structure with its respective correlation (Feldhusen & Grote, 2013).

Complementing the procedural V-model, there are product models to describe the detailed product information of technical systems during the development process (Ponn & Lindemann, 2011). They generally follow the principle of increasing concretization, like RUDE defines the 'model space of design' (*german trans. 'Modellraum des Konstruierens'),* consisting of four successive layers: requirements, functions, principle and shape (Rude, 1998). Being mechanical design oriented, RUDE'S model was refined for interdisciplinary product development with the Munich-Concretization-Model (MCM) (Ponn & Lindemann, 2011) and the model framework included in the mecPro (Eigner et al., 2017). They have similar layers, but extend towards mechatronic products while re-organizing the requirements layer. Furthermore BAUGHEY has introduced a development structure: Requirements-Functions-Logical-Physical, following the acronym RFLP (Baughey, 2011). This is also mentioned by EIGNER and KLEINER, while refining the left V-branch vertically using the acronym RFLP (Eigner et al., 2012; Kleiner & Kramer, 2013). The scheme can also be found in the aviation industry, as standards like the IEEE 1220-2005 give a hint about RFLP in the Systems Engineering Process (Guenov et al., 2016; IEEE Computer Society, 2005). All approaches follow the same pattern with a layer-structured composition: starting with the requirements, leading to functions, principle or logical solutions and finalized in the physical or technical realization. These layers generally help to structure and emphasize development stages.

Coming from RUDE's model in the classic mechanical product development and being advanced within mechatronic interdisciplinary product development, these models show maturity and are established in science. Yet their applicability in industry practice (especially automotive) cannot be observed. BAUGHEY points out that automotive development in practice covers mostly the perspectives requirements and physical components (Baughey, 2011). Especially for the complete vehicle concept development this is still the case (Busche, 2014; Felgenhauer, 2019; Pischinger & Seiffert, 2021). BAUGHEY suggests RFLP as a generic structure for the development, but gives no further insight on how to apply it. The approaches from EIGNER and KLEINER also lack practical relevance and do not describe the use of RFLP with highly complex products. This originates in the theoretical description with no operative model implementation, as models are state of the art to support the development in organizations along the development process (Eigner et al., 2014). The mecPro framework yet gives a first idea of how the implementation with Model-Based Systems Engineering (Eigner et al., 2017).

2.3 Model-Based Systems Engineering in automotive development

Model-Based Systems Engineering (MBSE) means 'the formalized application of modeling to support system requirements, design, analysis, verification and validation activities, starting in the conceptual development phase and continuing through the development [...]' (INCOSE, 2015). It is the model-based advancement of Systems Engineering (SE), a perspective and process concerning the development of (multidisciplinary) complex systems (Schulze, 2016).
SE promotes systems thinking, which enables 'a perspective of […] wholes and how the parts within those wholes interrelate'(INCOSE, 2015). It sets basic principles like developing a product from customer needs and functionality, thinking hierarchical, abstract and interdisciplinary, through an iterative process (Haberfellner et al., 2019). SE describes the fundamental principles, where MBSE applies them in a model-based manner with a central and common system model to document the multidisciplinary development (Hick et al., 2019). System models are the core of MBSE, as they represent a single source of truth and enable different engineering perspectives, while remaining consistent (Friedenthal et al., 2015). This ensures traceability during the development, covering all system elements, their cross-cutting relationships and related (development) information. System models consider three main aspects: requirements, behavior and architecture (Alt, 2012; Bajaj et al., 2016). The term 'architecture' is to be emphasized, being 'fundamental to any systems engineering undertaking' (Holt & Perry, 2014). Within (MB)SE architecture is defined as 'fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution' (ISO/IEC/IEEE 42010:2011). Accordingly these aspects are covered with the established Systems Modeling Language (SysML), which offers diagrams and modeling elements that represent different aspects of a system, further to be found in (Friedenthal et al., 2015).
In the automotive industry exist scattered applications of MBSE, mainly in the E/E or software-function development, like MAURER or BACH, that present approaches to develop autonomous driving functionality or E/E architecture (Bach et al., 2019; Maurer & Winner, 2013). System models are mostly used for smaller subsystems of new technologies, like steering (steer-by-wire) or assisted driving (Advanced Driver Assistance Systems). That is also what the SPES approach addresses, a framework for the development of software-intensive embedded systems (Pohl et al., 2012). The SPES presents a suitable methodology, covering the same architecture views functional, logical and technical seen in previous chapter of product theory.
Neither literature nor industrial observations show overall, complete vehicle and multidisciplinary MBSE and system architecture. Most approaches focus on certain domains, subsystems or technologies. There is no practice considering the complete vehicle, being holistic, including all engineering disciplines. Other than generic frameworks like the Automotive Architecture Framework (AAF) there are hardly practical and helpful approaches to be found (Broy et al., 2009).

3 Research objective

Looking at the inherent complexity and the earlier described market and technology challenges, cars are the best example for the need of SE thinking and approaches (Baughey, 2011; Gausemeier et al., 2013; Schulze et al., 2020). Literature and observations in industrial practice show a gap between the actual processes and theoretical SE. In practice, highly component-oriented processes and a domain-limited mindset can be observed, whereby the holistic system perspective is ignored (Baughey, 2011; Volkswagen AG, 2022). As vehicle concept development is located at the beginning of the V-model it has the ideal prerequisites to address this holistic system perspective. Further it is predestined due to the great leverage regarding influencing further development and realization of the product. Today vehicle concept development is focusing on mechanical and geometrical design, missing system and functional perspectives throughout all engineering disciplines. There is no overall methodology that addresses the complete vehicle from beginning of concept phase, on an overall system level. Therefore, this paper aims to transfer the theoretical and necessary approaches into practical context of vehicle concept development, by addressing the following question:

- How can SE be implemented within (today's) vehicle concept development?

Considering SE state of the art this also includes its model-based implementation (MBSE). As stated before, a core aspect of MBSE is architecture. This term was used multiple times showing different understandings in different science fields in Chapter 2. Vehicle concept development addresses architecture as the mechanical and geometrical structure of physical components. Product development theory further adds the functional structure, stating that product architecture is the correlation of physical parts and their functions. SE keeps the definition more general concerning systems elements and their relationships in between and underlines the important role of architecture. With respect to the shown challenges, it is necessary to determine a suitable adapted understanding of architecture in vehicle concept development, which combines the SE way of thinking with the existing classical concept development. Accordingly the following question is to be answered:

- How is architecture to be understood in (today's) vehicle concept development under the regard of SE?

Both questions aim for a new understanding of vehicle concept development in terms of Systems Engineering principles. By answering them, this publication is the introduction of an application-oriented methodology to enable structured and system-oriented architecture development within the concept phase of vehicle development.

4 Conceptual SE methodology for vehicle concept development

Based on the outlined state of the art, relevance and initial research questions the conceptual idea of a SE architecture methodology for vehicle concept development is described in the following. The requirements regarding the methodology are derived from literature and one year of practical observations within an established automotive manufacturer. Based on this, the basic idea of the methodology is presented afterwards.
Compared to the existing methods (see chapter 2), this methodology aims to cover both industrial need and scientific foundation, especially the adaptation on today's practice which is very component and mechanical driven. This methodology is not creating a completely new idea or structure, it is based on established and known theory. Yet the automotive industry is one of the most complex industries and the practical observations and discussions show that even experienced experts do not have the tangible methodical solutions for the architecture development.

4.1 Requirements for the methodology

To develop the methodology according to its purpose, the requirements and limitations are derived from the state of the art as well as observations and challenges from industry practice. The main aspect of the methodology is to follow SE principles: developing a product from customer needs and functionality with consistent traceability, requirements documentation, hierarchical, abstract, interdisciplinary and holistic thinking. The following specific requirements are to be considered in addition to SE:

- **Focus on the complete vehicle during concept phase** - The methodology concerns the complete vehicle architecture development with large complex functionalities or user scenarios that cannot be implemented with individual subsystems of the vehicle and thus require an overarching multidisciplinary development (e.g. the emerging automated driving).
- **Focus on interdisciplinary architecture development** - The methodology is intended to support and improve the development of cross-disciplinary and cross-sectional system architectures. It extends the existing geometric mechanical concept development and other approach that are dedicated to the development of E/E architectures, like SPES, MAURER or BACH. The focus is on the interdisciplinary interfaces and system conflicts, not on the technical detailing of certain engineering disciplines.
- **Support architecture development of a broad product portfolio** - The methodology is intended to address the platform systematic of vehicles, which is established for car manufacturers with a broad vehicle portfolio. It must cover the variants in the architecture of the portfolio and help to control the variance from the beginning. The methodology should make it possible to identify architecture-determining systems at an early stage and derive the effects on the other architecture elements/systems at an early stage.
- **Applicability within existing processes and product lines** – The methodology is to be pragmatically applied within existing vehicle development, product landscape and processes. The fundamentals of vehicle development are relatively consistent, therefor a 'green field' methodology is not practical.

4.2 Basic idea of the methodology

Based on these requirements, the idea of the methodology is explained in the following. The methodology follows a systematic core structure that originates in the theory of product models and SE (RUDE, LINDEMANN, EIGNER, BAUGHEY and BROY in chapter 2.2). It divides the vehicle concept development into four views: Requirements, Functional, Logical and Physical (RFLP).

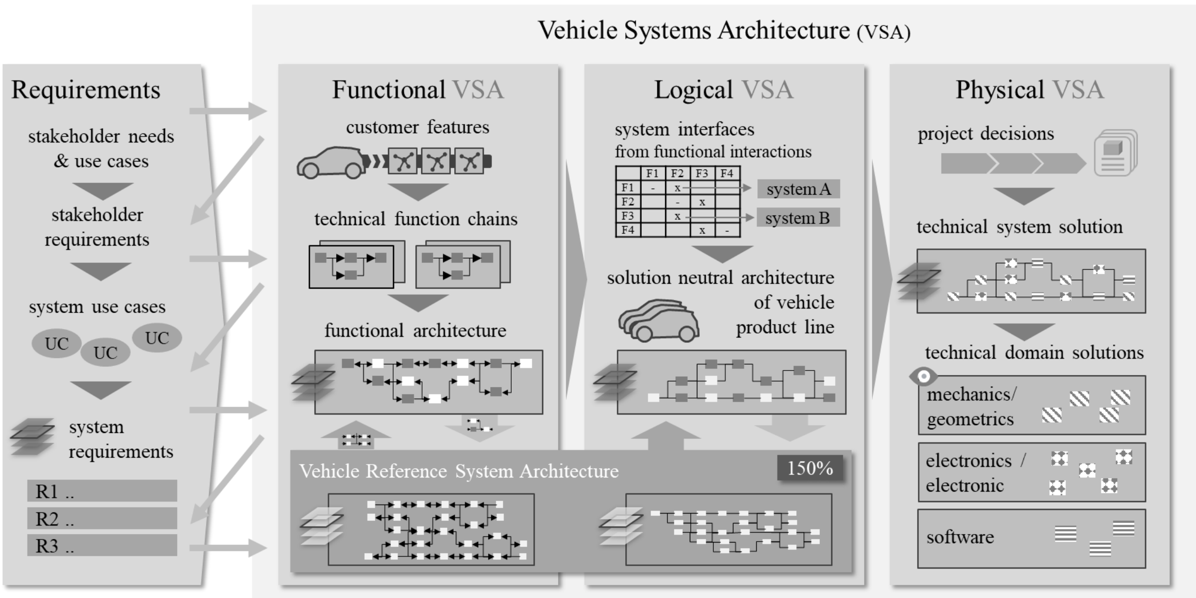

Figure 1. Methodology overview for Vehicle Systems Architecture concept development

These views aim to ensure structured consistency during concept development, as they represent the refining stages of development. The **requirements (R)** perspective presents the starting point of the methodology. It covers the stakeholder demands, business, legal and formal system requirements. F, L and P are the architectural perspectives which belong to and

introduce the '**Vehicle Systems Architecture**' (**VSA**) within the methodology and address the importance of architecture in the development (as outlined before). The VSA represents the new understanding of architecture in terms of SE oriented vehicle concept development. With F, L and P the existing understandings of architecture of product development theory, MBSE and vehicle concept development are combined. The **functional perspective (F)** is derived from R and includes two sides. First the **customer features**, that are functions the product offers to the customer, are 'marketable' and considered more at the overall vehicle level. On the other side **technical functions**, that are the 'enablers' of the customer functions, i.e. the functions that implement the customer function within technical systems. These technical systems are part of the more structural view, the **logical perspective (L)**. It represents the allocation of technical functions to systems leading to a solution neutral architecture. L is a pre-stage to the technical disciplines and thereby serves to improve the complexity handling and consistency of the development, especially within a broad product portfolio. Furthermore it is intended to establish a so-called **reference architecture** in the 'abstract' perspective of the F- and L-view. This reference should work similar to the known module and platform strategies, present a base architecture of multiple vehicle derivatives and thus leverage the synergies in a product portfolio. The final **physical perspective (P)** of the VSA addresses today existing concept architecture development and the engineering disciplines with its technical solutions. It includes the technical implementation of the architecture with system solutions and links to the disciplines of mechanics/geometry, E/E and software.
Furthermore the methodology is to be implemented in a **model-based** manner. Therefore a corresponding **meta-model,** that outlines the interrelationships and implementation of the methodology with the help of the modelling language SysML, is needed. The further detailing of these aspects and the complete methodology content is subject of ongoing research activities.

5 RFLP for vehicle concept development

RFLP as the core structure of the methodology suits as a procedural structure as well as architecture system views. As the V-model describes the solution specification in its left branch with the system concept, RFLP was seen analogue vertically in the left V-branch (Eigner et al., 2012; Kleiner & Kramer, 2013). This is an abstract interpretation as it doesn't support a hierarchical abstraction which is needed for the complexity of e.g. vehicles.

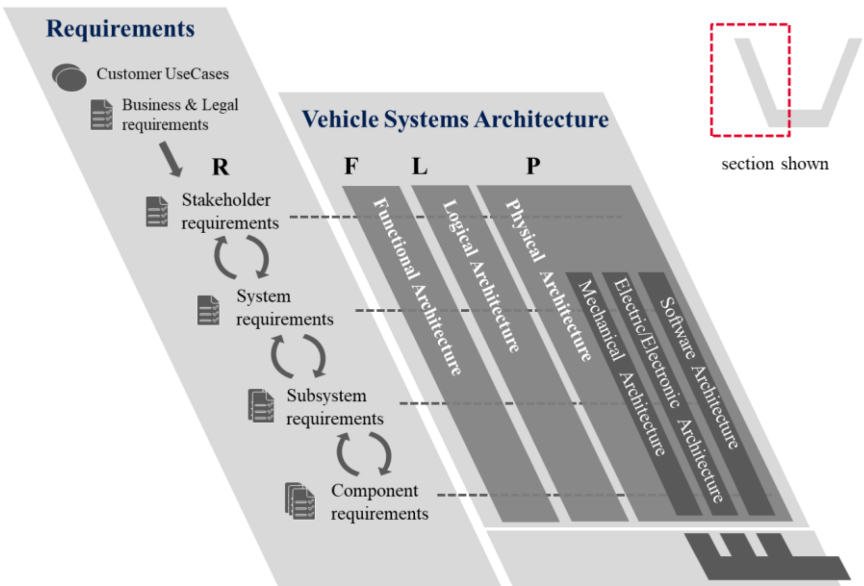

Figure 2. RFLP approach integrated in the left branch of the V-model

More suitable is the implementation of RFLP horizontally in the V (Esdras & Liscouët-Hanke, 2015) to benefit from combining the RFLP structure with hierarchical levels. This leads to the interpretation in Figure 2 of RFLP in the left branch of the V-model. RFLP strongly relies on an interconnection back and forth between requirements and the developed architecture. So far the dedicated iteration between requirements and architecture is not popular in automotive practice, at least not applied in a systematic manner. This raises an important aspect of SE and represents a key aspect of this methodology, to improve the zig-zag of Requirements and Architecture, preferably model-based.

5.1 Requirements perspective

Requirements are the starting point of each product development. They serve to document all stakeholder needs, boundary conditions and characteristics of the developed system (INCOSE, 2015). There exist two kinds of requirements: functional requirements, the text-based description of the system behavior and non-functional requirements, which are essential as they mostly influence the architecture design by specifying performance or quality of the systems (Alt, 2012). Non-functional requirements are mapped straight to the logical systems, to be available for all following elements. Furthermore it is possible that non-functional requirements on top level (e.g. regarding the drag coefficient c_w) lead to functional requirements in the following system levels, because a functionality (e.g. active aerodynamics) is needed to fulfill the targeted performance[1]. Almost every non-functional requirement is closely related to functional requirements, just as the drag coefficient is related to driving function (and its characteristic driving range).
In general requirements are meant to be solution independent (Alt, 2012; Eigner et al., 2014), yet with further specification of the requirements through the system levels, they are based on technical decisions made (in P) and therefore contain a certain solution reference.

5.2 Architecture perspective

The following describes how the perspectives of the VSA are to be understood and what aspects they cover in regard of vehicle concept development. Future research will examine each more detailed, including architectural artefacts and examples.

5.2.1 Functional perspective

Based on the (functional) requirements, the functions are derived, corresponding to the next step in product concretization (Ponn & Lindemann, 2011). Functions describe ‘what the system does [...] to fulfill objectives‘ (Haberfellner et al., 2019). On the top level the functional description is customer-oriented, to be called customer features (Pohl et al., 2012; Şahin, Köster et al., 2021). They are marketable overall functions the product provides to the user, e.g. ‚Active Cruise Control‘, ‚Park Assist‘ or automatic windscreen wipers. To implement a feature, several technical functions are necessary (e.g. detect environment, apply torque, set steering angle, detect rain drops, move wiper), which enable the feature in interaction, as a cause-effect-chain. Technical functions are implemented by systems and correspond to the definition by PONN, describing the relationship between input and output variables (Ponn & Lindemann, 2011). All technical functions of the complete vehicle form the functional architecture, showing all interrelations and dependencies of the system‘s functionality. Usually technical functions are

[1] The aerodynamic performance of an electric vehicle in the volume market is mostly driven be energy efficiency. A low drag coefficient c_w contributes to longer range of electric vehicles and therefore improves a non-functional system characteristic. Based on the requirement (‘have a low drag coefficient to reduce energy consumption’), functional requirements can be derived (e.g. active aerodynamics, improving the areo-form with moving spoiler or rear diffuser at certain speeds on the highway).

not observable by the user. The feature instead is directly observable. This differentiation enables a more customer-oriented development and supports the traceability, from an abstract overall product level down to the technical functionality of the components. Thereby it is apparent which functions are responsible for the feature.

5.2.2 Logical perspective

The technical functions are allocated and clustered into logical systems according to selected architecture criteria (criteria see Haberfellner et al., 2019). Through the interactions of the functions, the interfaces and interrelations of the allocated systems are derived, forming the L-architecture. The logical perspective is (primarily) solution-neutral and does not yet define how (mechanical, electrical, software) the realization takes place. This also allows to map generally valid, non-functional requirements (e.g. legal demands) to the logical systems and make them traceable for all subsequent systems and P realizations.

The L-view is a tool to improve the handling of complexity. It enhances the architecture in respect of its functional performance, as the L-systems are optimally tailored according to functionality. Being solution independent, it supports highly varying products, matching the platform and modular strategy of automotive manufacturers. The L-view serves as a basis for product variants later defined in P. This way a L-architecture can be defined platform-wide and re-used in all subsequent car variants (derivatives), as a detailed solution remains open. For example the logical architecture can be defined for an electric vehicle (platform), with basic architecture systems, like energy storage, thermal management or engine. The differences of specific vehicles happen in P (of e.g. storage capacity, power, material or dimensioning of the thermal management).

5.2.3 Physical-technical perspective

The P-level is the final technical implementation of the developed system. Tangible decisions are made about which technical solution is chosen to implement the functions and (logical) systems. In established, existing industries (like automotive development) the P-perspective has a high relevance. Here already existing realization-related engineering information can be integrated. For example is the energy storage system still solution-independent in L and now becomes an electric 400V Li-Ion Battery in P. At the same time, the interfaces, which were formulated in a more solution-independent way in L, are specified. For example, an information becomes a data bus signal, and thermal energy becomes a coolant through specified pipes.

5.3 RFLP framework for system and domain models

RFLP describe the methodical refinement, developing a system from requirements to a physical solution. In the same manner RFLP are consistent system (architecture) perspectives, like FLP as perspectives on the VSA. Both aspects do not contradict each other, but for consistent perspectives an iterative development via RFLP is necessary. This means that solution decisions in P may have retroactive effects on the F-view, which then have to be added afterwards. Figure 3 addresses the R to P development step horizontally. The iterative part happens via the vertical levels on the downstream. Here, the views are detailed and thus contain the decisions of the higher level, in the sense of Black Box/White Box development (Pohl et al., 2012; Şahin, Raulf et al., 2021). Each level is connected with the previous and following level. Decisions in level n affect the following level n+1 and are, according to the hierarchical SE principle, more specific and detailed. Increasing solution dependence (visualized in Figure 3 ‘solution specific’) includes the technical principle and project decision P_{sn} (e.g. electric storage or fuel tank) and the involved engineering discipline with its domain-specific details P_{dn} (e.g. geometry and E/E architecture of the electric storage). Today’s vehicle concept development is located in the P_{dn}, directly linked to the project decisions, having a strong role during the whole procedure.

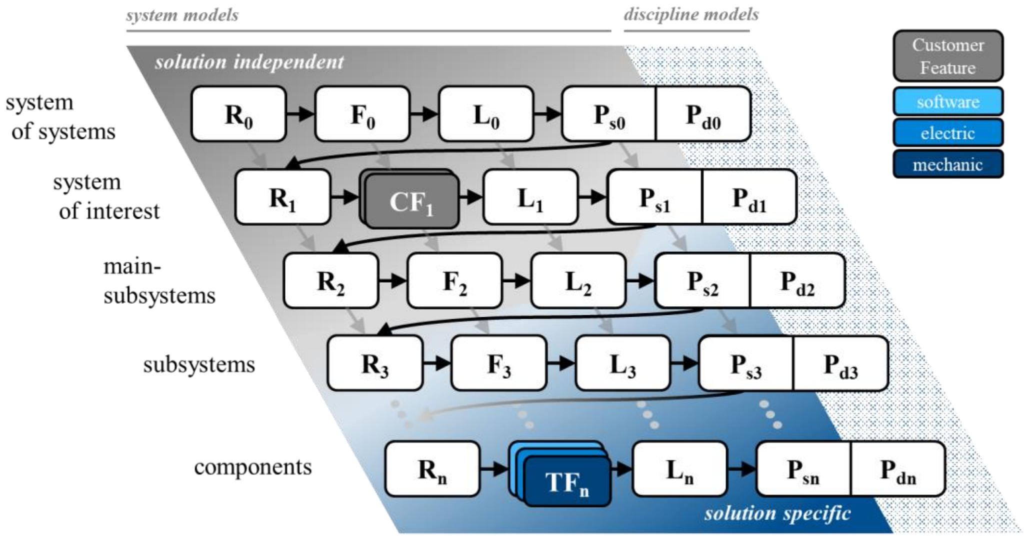

Figure 3. Hierarchical RFLP framework with system and discipline models

As every system level includes a decision in P that leads to the next level requirements, the solution neutral R_n, F_n and L_n are depending on the P_{n-1} decisions made before. As a consequence of the hierarchical system levels, F and L can be depending on technical (solution) decisions and get solution specific in lower system levels. They depend on the previous derived requirements which are based on the technical decisions made. For example 'cool the system' is a solution independent function to the point where to decide technically if it is realized e.g. by air or fluid cooling. Each decision will consequently affect the subsequent levels and is followed by a different architecture. The P_{sn} decision 'fluid cooling' will probably need a circular flow and functions like 'move fluid', 'guide fluid', 'exchange heat' that lead to solutions like pumps, pipes and heat exchangers, while air cooled systems can be realized as open, non-circular system differently. The detailed geometrical and electrical design of e.g. the radiator and cooling cycle happens in P_{dn}. The functions depend on the previous technical decisions and thereby lose solution independence. This specification (even on higher levels) and narrowing of the solution space is a necessary part of the project progress and mandatory for successful structured architecture development. Thus, domain aspects can already occur on high system levels. This is an essential aspect of this RFLP framework, especially in the context of today's vehicle concept development, since the geometric basis of the vehicle must be developed and taken into account at a high level of abstraction.

6 Conclusion and further research

The implementation of SE within vehicle concept development requires a new methodology. It follows the core structure requirements, functions, logical and physical (RFLP), which is derived from state of the art research. F, L and P belong to the architectural perspectives and thereby introduce the Vehicle Systems Architecture. The VSA represents the new understanding of architecture development in vehicle concept phase and is aligned with the RFLP methodical model. Located in the left branch of the V-model, RFLP aims to consciously document and guide the development processes of the architecture. It improves the awareness of the customer requirements, function-oriented development through multidisciplinary systems and enables consistency and traceability e.g. for legal proofs through the whole architecture system development. The goal is to make the complexity in the development process more controllable and improve the management of changes and indirect dependencies of the functions, systems and components.

This contribution introduces the basic idea and structure of the methodology according to the RFLP framework in the context of vehicle architecture concept development. Future research will be on the detailing of each VSA views FLP, as well as the transition from view to view. Further, the interplay between the existing concept development, regarding geometric design and general vehicle characteristics, with these new approaches is to be examined. This includes the work with large product portfolios, platforms and the idea of a reference architecture. The whole methodology is to be implemented with a system model in the center of the development activities. Likewise the system model implementation including a meta model with SysML will be defined within future research activities.

Disclaimer:
The results, opinions and conclusions expressed in this paper are not necessarily those of Volkswagen AG.

7 References

Alt, O. (2012). *Modellbasierte Systementwicklung mit SysML*. Carl Hanser Verlag GmbH Co KG.

Anderl, R., Eigner, M., Sendler, U., & Stark, R. (2012). *Smart Engineering*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29372-6

Bach, J., Otten, S., & Sax, E. (2019). Classification of Automotive Electric/Electronic Features and the Consequent Hierarchization of the Logical System Architecture. Advance online publication. https://doi.org/10.1007/978-3-030-02907-4_12

Bajaj, M., Cole, B., & Zwemer, D. (2016). Architecture To Geometry - Integrating System Models With Mechanical Design. Advance online publication. https://doi.org/10.2514/6.2016-5470

Baughey, K. (2011). Functional and Logical Structures: A Systems Engineering Approach. In *SAE Technical Paper Series, SAE Technical Paper Series.* SAE International400 Commonwealth Drive, Warrendale, PA, United States. https://doi.org/10.4271/2011-01-0517

Broy, M., Gleirscher, M., Kluge, P., Krenzer, W., Merenda, S., & Wild, D. (2009). *Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description.*

Busche, I. (2014). *Ein Beitrag zur optimierten Konzeptauslegung von Fahrzeugen im Bereich der Elektromobilität* [Dissertation, Otto-von-Guericke-Universität Magdeburg, Magdeburg]. GBV Gemeinsamer Bibliotheksverbund.

Dumitrescu, R., Albers, A., Riedel, O., Stark, R., & Gausemeier, J. (Eds.). (2021). *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft, Ein Beitrag zum Advanced Systems Engineering.*

Eigner, M., Dickopf, T., Schneider, M., & Schulte, T. (2017). MecPro2- Holistic Concept for the Model-Based Development of Cybertronic Systems.

Eigner, M., Gilz, T., & Zafirov, R. (2012). Interdisciplinary Product Development - Model Based Systems Engineering.

Eigner, M., Roubanov, D., & Zafirov, R. (Eds.). (2014). *Modellbasierte virtuelle Produktentwicklung*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43816-9

Esdras, G., & Liscouët-Hanke, S. (2015). Development of Core Functions for Aircraft Conceptual Design: Methodology and Results.

Feldhusen, J., & Grote, K.-H. (Eds.). (2013). *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. Springer Berlin Heidelberg.

Felgenhauer, M. (2019). *Automated Development of Modular Systems for the Vehicle Front of Passenger Cars* [Dissertation]. Technische Universität München.

Friedenthal, S., Moore, A., & Steiner, R. (Eds.). (2015). *A Practical Guide to SysML: The Systems Modeling Language*. Elsevier.

Gausemeier, J., Dumitrescu, C., & Steffen, D. (2013). Systems Engineering in der Industriellen Praxis.

Guenov, M., Molina-Cristobal, A., Voloshin, V., Riaz, A., van Heerden, A. S., Sharma, S., Cuiller, C., & Giese, T. (2016). Aircraft Systems Architecting - a Functional-Logical Domain Perspective. Advance online publication. https://doi.org/10.2514/6.2016-3143

Gusig, L.-O., & Kruse, A. (Eds.). (2010). *Fahrzeugentwicklung im Automobilbau: Aktuelle Werkzeuge für den Praxiseinsatz*. Hanser Verlag. https://doi.org/10.3139/9783446424685

Haberfellner, R., Weck, O. de, Fricke, E., & Vössner, S. (Eds.). (2019). *Systems Engineering*. Springer. https://doi.org/10.1007/978-3-030-13431-0

Hick, H., Bajzek, M., & Faustmann, C. (2019). Definition of a system model for model-based development. *SN Applied Sciences*, *1*(9), 15. https://doi.org/10.1007/s42452-019-1069-0

Holt, J., & Perry, S. (Eds.). (2014). *SysML for Systems Engineering*. IET. https://doi.org/10.1049/PBPC007E

IEEE Computer Society. (2005). *IEEE Std 1220-2005 Standard for Application and Management of the Systems Engineering Process*. http://ieeexplore.ieee.org/servlet/opac?punumber=10106

INCOSE. (2015). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities* (4th ed.). Wiley.

Kleiner, S., & Kramer, C. (2013). Model Based Design with Systems Engineering Based on RFLP Using V6. In M. Abramovici & R. Stark (Eds.), *Smart Product Engineering* (pp. 93–102). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30817-8_10

Maurer, M., & Winner, H. (Eds.). (2013). *Automotive Systems Engineering*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36455-6

McKinsey & Company (Ed.). (Juli 2019). *Automotive software and electronics 2030: Mapping the sector's future landscape*. https://www.mckinsey.de/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030

Michels, J. S. (2016). Vom Kunden zum Lastenheft – Systems Engineering in den frühen Phasen der Entwicklung intelligenter technischer Systeme. In T. Abele (Ed.), *Die frühe Phase des Innovationsprozesses* (pp. 163–189). Springer Fachmedien Wiesbaden.

Pischinger, S., & Seiffert, U. (Eds.). (2021). *Vieweg Handbuch Kraftfahrzeugtechnik*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-25557-2

Pohl, K., Hönninger, H., Achatz, R., & Broy, M. (2012). *Model-Based Engineering of Embedded Systems*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-34614-9

Ponn, J., & Lindemann, U. (2011). *Konzeptentwicklung und Gestaltung technischer Produkte*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20580-4

Rude, S. (1998). *Wissensbasiertes Konstruieren*. Zugl.: Karlsruhe, Univ., Habil.-Schr., 1998 (Als Ms. gedr). *Berichte aus dem Maschinenbau*. Shaker.

Şahin, T., Köster, L., Huth, T., & Vietor, T. (2021). How to Upgrade Vehicles? Release Planning in the Automotive Industry. In M. Bargende, H.-C. Reuss, & A. Wagner (Eds.), *Proceedings. 21. Internationales Stuttgarter Symposium.* Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-33521-2_40

Şahin, T., Raulf, C., Kizgin, V., Huth, T., & Vietor, T. (2021). A Cross-domain System Architecture Model of Dynamically Configurable Autonomous Vehicles. In M. Bargende, H.-C. Reuss, & A. Wagner (Eds.), *Proceedings. 21. Internationales Stuttgarter Symposium.* Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-33521-2_40

Schulze, S.-O. (2016). Systems Engineering. In U. Lindemann (Ed.), *Handbuch Produktentwicklung* (pp. 153–184). Hanser.

Schulze, S.-O., Steffen, D., Wibbing, P., & Wigger, T. (2020). *Digitalisierung der Produktentstehung - Die Automobilindustrie im Umbruch.*

ISO/IEC/IEEE 42010:2011 (2011). *Systems and software engineering — Architecture description*.

Toepfer, F., & Naumann, T. (2016). Management of Vehicle Architecture Parameters. *INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016*.

Verein Deutscher Ingenieure (2019). VDI 2221 - Design of technical products and systems.

Verein Deutscher Ingenieure (2020). VDI/VDE 2206 - Development of mechatronic and cyber-physical systems.

Volkswagen AG. (2022, March 3). *Volkswagen richtet Technische Entwicklung neu aus: mehr Tempo bei Produktzyklen und digitalen Angeboten* [Press release]. Wolfsburg. https://www.volkswagen-newsroom.com/de/pressemitteilungen/volkswagen-richtet-technische-entwicklung-neu-aus-mehr-tempo-bei-produktzyklen-und-digitalen-angeboten-7768