New Methodologies for Automotive PLM by Integrating 3D CAD and Virtual Reality into Function-oriented Development

Dem Fachbereich 3 (Mathematik und Informatik)
der Universität Bremen
eingereichte

Dissertation

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Dipl.-Inf. Moritz Cohrs

Referenten der Arbeit:	Prof. Dr. Gabriel Zachmann
	Prof. Dr. Franz-Erich Wolter

Abstract

Today, a primary challenge within the automotive industry is the high and further increasing complexity of modern vehicles, caused by the growing numbers of vehicle functions, electronics and software. Considering this challenge, the approach of a function-oriented development extends the traditional component-oriented development by focusing an interdisciplinary development of vehicle functions as mechatronic and cyber-physical systems and it is an important measure to master the increasing product and development complexity. In addition, technologies like virtual reality, computer-aided design (CAD) and virtual prototyping offer largely established tools for the automotive industry in order to handle different challenges across product lifecycle management. So far, however, the promising potentials of 3D virtual reality methods have not yet been evaluated in the particular context of an automotive function-oriented development. Therefore, this research focuses on the question if and how 3D virtual reality methods can improve relevant workflows and generally streamline a function-oriented development. To address this research goal, a prototypical implementation is developed which is based on a consistent integration of vehicle function architectural data with 3D CAD data. In particular, a primary contribution of this thesis is the research and development of novel, function-oriented 3D methods which are enabled by this kind of data integration. Based on the prototypical implementation, these methods are designed, developed and evaluated. With the application on different automotive use cases it is shown that these novel 3D methods provide a significant benefit for function-oriented development and comparable approaches to systems engineering. Moreover, proposals are provided on how these methods can be integrated into prospective 3D tools in order to solve relevant function-oriented tasks much more efficiently in the future. In addition, user studies are conducted to further evaluate the novel 3D methods and to identify improvement potentials and areas for future work.

Disclaimer

Publications about the content of this work require the written consent of Volkswagen Aktiengesellschaft. The results, opinions and conclusions expressed in this thesis are not necessarily those of Volkswagen Aktiengesellschaft.

Product-related data and artifacts used in this thesis which are marked as hypothetical feature the same statistical properties and characteristics as real industrial data and artifacts.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	Three-dimensional
API	Application Programming Interface
AR	Augmented Reality
CAD	Computer-aided Design
CAE	Computer-aided Engineering
CAN	Controller Area Network
CAx	Computer-aided Technologies
CLD	Chromatic Light Deflector
CMC	Computer-mediated Communication
CPS	Cyber-physical System(s)
CSCW	Computer Supported Cooperative Work
DLP	Digital Light Processing
DMS	Data Management System
DMU	Digital Mock-up
ECU	Electronic Control Unit
FDMU	Functional Digital Mock-up
FMU	Functional Mock-up
GFLT	Geometric Function Localization Task
HMD	Head-mounted Display
HMI	Human-machine Interface
ISO	International Organization for Standardization
IT	Information Technology
JT	Jupiter Tessellation
KBE	Knowledge-based Engineering
LOD	Level of Detail
MR	Mixed Reality
NURBS	Non-Uniform Rational B-Spline
OEM	Original Equipment Manufacturer
PLM	Product Lifecycle Management
PMU	Physical Mock-up
SHP	Smart Hybrid Prototyping
SR	Simulated Reality
SysML	Systems Modeling Language
VDI	Verband deutscher Ingenieure
VE	Virtual Environment
VP	Virtual Prototyping, Virtual Prototype
VR	Virtual Reality
VT	Virtual Technologies
XML	Extensible Markup Language
XOR	Exclusive OR (electronic logic gate)

1 Introduction

This chapter introduces current challenges in automotive product development and lifecycle management which have provided the motivation for this thesis. In addition, the primary research question of this thesis is phrased and an overview of the scientific contributions and the thesis structure is outlined.

1.1 Motivation

Today, the automotive industry has to face a considerable diversity of challenges. Globalization has caused strong international competition and has significantly extended the call for innovation and the overall scope of markets and customer requirements. Even within single markets, increasing customer requirements lead to significant demands for mass product customization and individualization. Moreover, there are more and more requirements due to legal, economic and environmental constraints. At the same time, product lifetime cycles have shortened and time to market is a crucial aspect for staying competitive and for exploiting innovation values.

One of the most significant challenges in the automotive industry is the high degree of product complexity of modern vehicles [1-3]. This complexity is caused by the ever increasing amount of vehicle functions, electronics and software. The increasing product complexity poses a major challenge for product development, quality assurance, customer service and many other related domains of the product life cycle management (PLM). A promising measure to master complexity in development cycles is the relatively new *function-oriented* approach of development. A function-oriented development is an expression of systems engineering and extends a traditional component-oriented development by focusing an interdisciplinary development of vehicle functions as mechatronic systems, which helps to handle the increasing complexity in current automotive development [1], [4,5]. Considering the current state of the art in automotive development, a function-oriented development or comparable systematic approaches to development are likely the most important measures to handle the high and further increasing complexity that has to be faced in this industry.

A well-known technology that assists in mastering automotive complexity is the utilization of computer-aided design (CAD), which is described by [6] as the use of computer systems to aid in the creation, modification, analysis and optimization of a design. A related area in the field of computer-aided tools is the field of applied virtual reality (VR) methods, respectively virtual technologies, which are fairly well-established technologies in the automotive industry as well. Such methods usually involve different types of 3D visualization of product data, and, if needed, also the element of immersive user interaction. Virtual reality methods are used in multiple areas of the automotive product lifecycle management, including but not limited to design, analysis, simulation and visualization. A typical automotive application of virtual reality methods is a digital mock-up. A digital mock-up is a 3D CAD model of a vehicle, respectively a virtual product prototype, which, for example, can be used for geometric analyses and assembly validations at early stages of development.

Both the function-oriented development approach and 3D CAD and virtual reality technologies provide support to better handle the increasing automotive product complexity. However, an interdisciplinary conjunction of these fields is not consequently established and current function-oriented development processes do not yet exploit the thorough capabilities of virtual reality methods. Also, a recent evaluation of current and upcoming technology underlines that current development processes do not fulfil the new requirements of the increasing complexity in the automotive industry and the authors highlight the necessity for new approaches for integration of geometric data with enhanced product information in order to increase the efficiency of virtual product development processes [7]. Moreover, [8] presented an analysis of literature from 1992 to 2014 related to virtual reality in the manufacturing industries which does not include any research on integrating function-oriented systems engineering with 3D CAD and VR. So, to the best of my knowledge, the promising potentials of 3D virtual reality methods have not yet been evaluated in the particular context of an automotive function-oriented development and its specific challenges. Moreover, there is no research on the novel possibilities and benefits which are enabled by consequently integrating and combining PLM artifacts and data of these two heterogeneous domains. Consequently, this has been an initial motivation for the thematic focus of this thesis as it is illustrated in Figure 1.

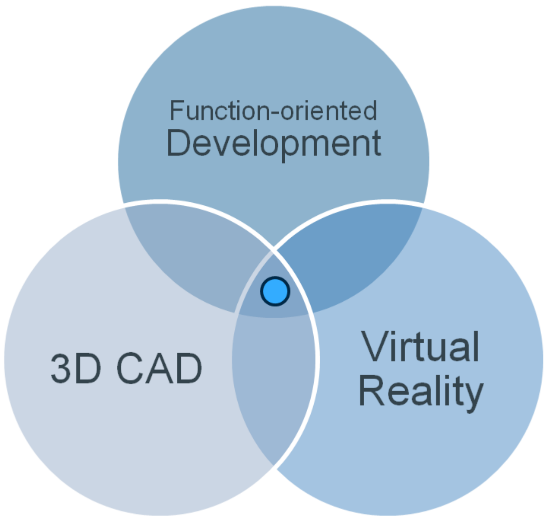

Figure 1 – The research work of this thesis focuses an integration of 3D CAD and virtual reality methods with function-oriented development

1.2 Research Question and Contributions

[9] have evaluated CAD systems regarding engineering design and analysis and point to the current issue of *domain isolation: "[…] advances transcending discipline boundaries have been disappointingly scarce. The domain isolation has had a significant impact on CAD to the point where engineers now spend a large portion of their time wrestling to overcome the resulting trans-area barriers."*

The focus of this thesis suggests an approach to overcome such a *domain isolation* and its feasibility is demonstrated on the example of connecting the domains of 3D CAD/VR and function-oriented development. In particular, this thesis focuses on the research question if and how 3D virtual reality methods can improve relevant automotive development workflows and generally streamline a function-oriented development and comparable approaches to systems engineering. In order to address this question, the work of this thesis primarily implies development, research and evaluation of novel function-oriented 3D methodologies.

In particular, this research work proposes a novel data integration concept which combines automotive function architecture data with geometric 3D CAD models as illustrated in Figure 2.

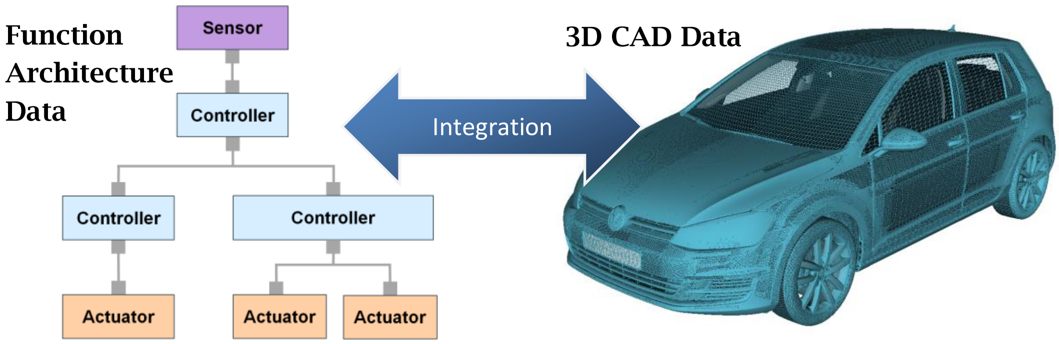

Figure 2 – The integration concept focuses a conjunction of automotive function architecture data with 3D CAD data

This interdisciplinary approach of data integration exploits the potentials of connecting two heterogeneous domains of automotive development as shown in Figure 3.

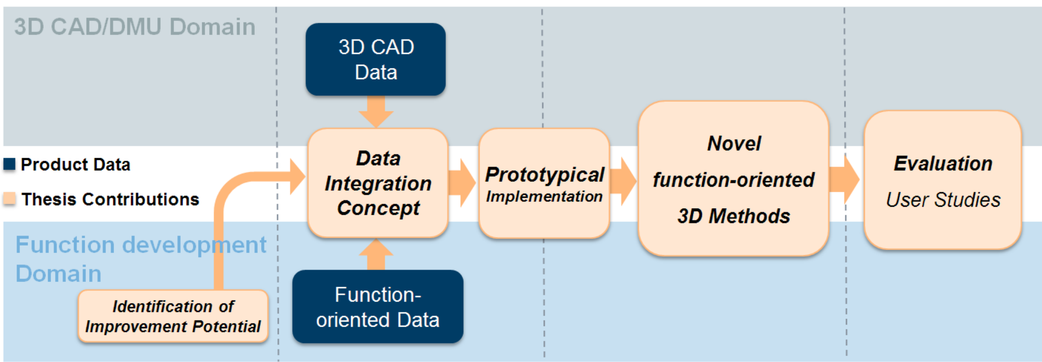

Figure 3 – This thesis connects two heterogeneous domains of automotive development in order to provide novel, beneficial 3D methods for function-oriented product lifecycle management

The above data integration concept is prototypically implemented in order to prove the feasibility of the concept and to enable development of novel function-oriented 3D methodologies. Consequently, the developed methods are evaluated regarding their capabilities for visualization, analysis and communication of function-oriented data. In addition, the developed methods are applied to a selection of automotive use cases across multiple PLM areas such as development, validation and service to evaluate their particular benefits among different use cases. Eventually, a selection of user studies is conducted to further evaluate the function-oriented 3D methods in terms of task-performance, usability and improvement potentials, and to compare them to traditional methods.

As follows I provide a short outline of the structural composition and particular contributions of the thesis.

Chapter 2 includes introductions and definitions of relevant subjects, such as technologies, research fields and data structures. Thus, theoretic fundamentals are described and the understanding and interpretation of related terms within the context of this thesis is clarified.

The following chapter (Chapter 3) provides an overall introduction to the approach of function-oriented development in the automotive industry. First, current challenges are discussed in order to provide a more detailed background for the motivation of this thesis. Then, an overview of the function-oriented approach to development is provided and its basic principles and benefits are discussed. Moreover, examples of typical vehicle functions are presented and the method of function-architecture design is introduced. Finally, the task of spatially localizing vehicle function elements is defined and I argue that current non-3D and non-function-oriented methods are not able to efficiently solve this task which highlights a current methodical gap as well as the relevance of this research work.

Chapter 4 explores the subject of virtual reality (VR) and related 3D CAD research. This chapter begins with an introduction and review of different VR definitions in order to identify and define the position of virtual reality within this thesis. Also, a short overview of technological aspects is provided and I present the results of my evaluation of an autostereoscopic 3D display for collaborative design reviews. In addition, a selection of multiple VR applications is discussed, including a broad selection of different use cases and related fields of research. Moreover, virtual reality is discussed as being a subject of new media in order to define its suitability and specific capabilities to particularly support a function-oriented development. Eventually, related research works in the field of interdisciplinary approaches to CAD, VR and virtual prototyping are discussed to elaborate current research gaps as well as trends and potentials. Hereby I further argue for the necessity of the interdisciplinary integration approach that I have chosen for this thesis.

Beginning with chapter 5, I present the thesis main contributions. Therefore, I first propose a concept for an interdisciplinary and consistent integration of automotive function architecture data with 3D CAD data. This work includes an identification and definition of basic requirements which are necessary for a consistent mapping between function-oriented data and 3D geometries. In addition, I present a self-developed system-independent XML-based data format for function architecture data which is able to simplify the data integration process and I show that it can be implemented with existing systems. Two technical variants for implementation of the data integration are proposed and compared. Then, I present my prototypical implementation in order to prove the overall feasibility of the integration concept. Summarizing, this implementation provides the basis for the further contributions and it enables the development and research of novel function-oriented 3D methodologies based on the prototype.

Chapter 6 seamlessly ties up to the previous chapter. Based upon the prototypical implementation, I develop and present a novel, function-oriented 3D methodology. In this context, I elaborate and discuss different examples on how this methodology can be potentially integrated into next-gen 3D tools to better support a function-oriented development. Moreover, I demonstrate benefits of these novel 3D methods by applying them on multiple automotive use cases in different areas like development, validation, maintenance and service.

Chapter 7 aims at further evaluating and assessing the function-oriented 3D methodology with user studies in order to both highlight benefits and identify areas for further improvement. Therefore, different user studies are conducted which focus on task performance, efficiency, solution correctness, usability, comparison of traditional methods vs. the function-oriented 3D methods and manual vs. automated performance.

The final chapter provides a summary of the thesis contributions and conclusions, as well as an outlook and discussion of potential future fields of research.

Summarizing, an overview of the primary thesis contributions is listed as follows:

- Elaboration of suitability and characteristics of virtual reality, considered as an interdisciplinary tool and multidimensional medium, in order to particularly support a function-oriented development.
- Development of an interdisciplinary data integration concept, including elaboration and definition of basic requirements for a consistent integration of 3D CAD models with function-oriented data.
- Prototypical implementation of the data integration concept to prove its feasibility and to enable research and development of novel methodologies based on the prototype.
- Development and research of novel and beneficial function-oriented 3D methods with the goal to streamline function-oriented workflows, incl. elaboration of benefits in automotive use cases.
- Evaluation of the function-oriented 3D methods with user studies, focusing on task-performance and usability, including comparison with conventional methods
- Identification of potentials for future work.

2 Overview of CAD, VR, related Subjects and Data Structures

This chapter briefly explores subjects related to computer-aided design (CAD), virtual prototyping (VP) and virtual reality (VR), including related research fields, technologies, development approaches and data structures. It is supposed to provide a basic theoretical groundwork for the thesis contributions in the later chapters. While this chapter also shortly introduces VR, the subject of virtual reality is primarily discussed in the dedicated Chapter 4.

2.1 Terminology

In the thematic fields of CAD, virtual reality and virtual prototyping, there are many recognized terms. Some of these terms might appear similar or even interchangeable. However, particular terms still evince differences so that a clear definition and separation is necessary to provide a consistent understanding of these fundamentals within this thesis.

2.1.1 Computer-aided Design (CAD)

Computer-aided design (CAD) is the use of computer systems to assist in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing [6]. CAD is a specific derivation of the more generic field of *computer-aided technologies* (CAx), which also includes other derivations like *computer-aided manufacturing* (CAM) or *computer-aided engineering* (CAE). Typical reference literature that focuses on CAx in the engineering domain is provided by [10]. In the automotive industry, CAD is a well-established technology that is used in multiple areas of the PLM. In this thesis, a particular focus is set on CAD for geometric design for 3D product prototypes. For further reading on CAD multiple of literature is available such as [11-13].

2.1.2 Virtual Prototyping (VP)

Basically, *virtual prototyping* (VP) is the use of software to validate a design before committing to make a physical prototype. For example, virtual prototyping is described in literature as the utilization of realistic product models for design and functionality analysis at early PLM stages [14]. Consequently, a *virtual prototype* is a digital representation of an industrial product that may include any product properties. For example, [15] defines a virtual prototype as a computer simulation of a physical product that can be visualized, analyzed, and tested from concerned product life-cycle aspects such as design/engineering, manufacturing, service, and recycling as if on a real physical model. [16] noted that virtual prototyping was pioneered and initially adopted by the automotive and aerospace industries. Moreover, the authors provide a survey of virtual prototyping techniques for mechanical product development and discuss related benefits and research challenges. Figure 4 shows an example of a virtual prototype, illustrating a geometric 3D CAD model of a vehicle.

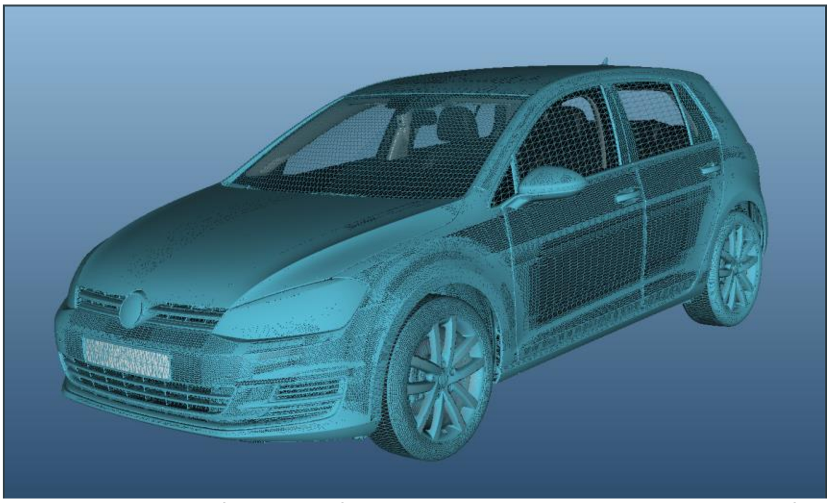

Figure 4 – A Virtual prototype of a vehicle featuring a geometric 3D representation of the product

It can be noted that not all aspects which are claimed in some literature definitions are necessarily fulfilled by every virtual prototype. So, the scope of which product properties and behaviors are represented and/or simulated by a virtual prototype usually can be dependent on the specific use case. For example, a virtual prototype may exclusively include geometric 3D data because it is supposed to be used for assembly analysis. However, this virtual prototype might not necessarily include additional product information, so, for example, it will not be usable for physical simulation as well. [17] discussed the application of virtual reality to virtual prototyping in order to verify assembly and maintenance engineering processes and the authors indicated the potentials of combining virtual reality and virtual prototyping. Additional examples of virtual prototyping in the automotive industries are discussed by [18] including problems, solutions and future directions.

2.1.3 Digital Mock-up (DMU)

A *digital mock-up* (DMU) is considered a specific utilization of virtual prototyping and describes the use of virtual prototypes for geometric analyses like assembly studies and design reviews. Before appropriate CAD and DMU tools were available, all prototypes had to be built from real world materials like cardboard or wood. These traditional types of prototypes are called *physical mock-ups* (PMU). In comparison to a physical mock-up, a digital mock-up usually is significantly more time and cost-efficient. In some literature like [15], a digital mock-up (DMU) has the same meaning as a virtual prototype. In this thesis, however, the terms of virtual prototypes and digital mock-ups are separated to keep definitions explicit and close to the understanding in the automotive industry where DMU focuses on geometric analysis of 3D models and virtual prototyping describes a more generic application of virtual product models in the PLM. An example of DMU data is shown in Figure 5. Further read is to be found in literature such as [19] and [20].

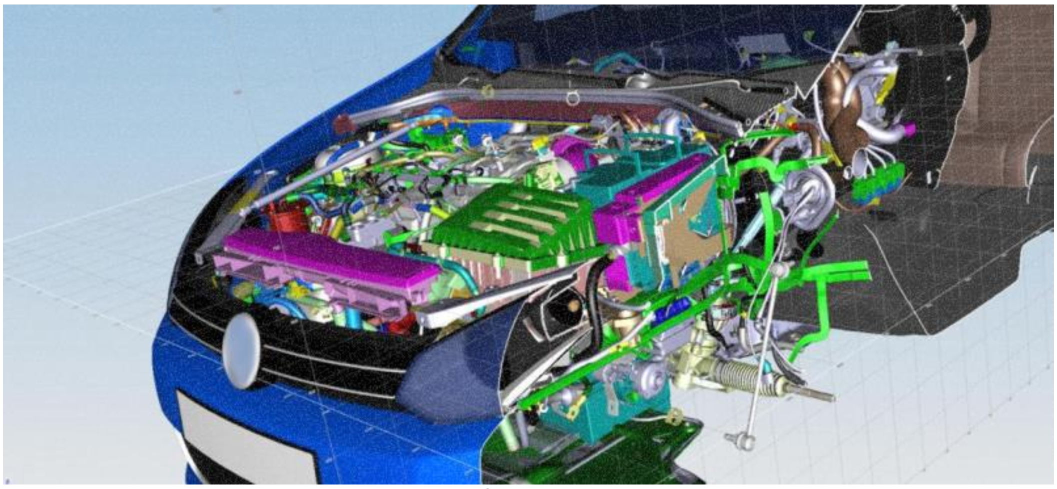

Figure 5 – Digital mock-ups enable analyses of virtual product prototypes at early PLM stages

2.1.4 Functional Mock-up (FMU/FDMU)

A *functional (digital) mock-up* (FMU/FDMU) is a particular application of virtual prototyping which enhances traditional digital mock-up by integrating numerical simulation models with CAD data to enable functional simulation of product properties [21]. The primary aim of a functional mock-up is the simulation of vehicle properties, functions and behavior upon a virtual prototype [22]. Therefore, simulation models are integrated with virtual prototypes using specific tools and modeling languages (e.g. SysML, VHDL-AMS, MATLAB, Modelica). For example, focal points can be in the simulation of tolerances, electrics, hydraulics, mechatronic components and manufacturing processes and facilities. A functional mock-up increases synergy effects between data acquisition, mechanic construction, functional product development, behavior simulation and integration. [23] pointed that a coupling of associated software and simulation tools is imperative for early functional integration of virtual mechatronic products. [24] assume that future work will aim to develop simulation toolkits that enable the verification of all essential vehicle functions in the concept phase.

[25] reported a lack in independent standards for model exchange and co-simulation and therefore introduced the Functional Mockup Interface (FMI). This tool-independent standard focuses on the exchange of dynamic models and on streamlining co-simulation to improve collaboration between suppliers and automotive companies. Having been started under the MODELISAR project, the FMI has arrived at the 2.0 standard (beta) [26] and is continued under the Modelica Association Project. Moreover, differences, advantages and disadvantages of both concepts, FMU and FMI, have been explored by [27]. The authors highlight the FMU focus on interactive 3D visualization including functional simulation and the FMI focus on efficient co-simulation and model exchange. By the proposal of three options, the authors argue that these concepts can be complementary combined for comprehensive investigation of multi physical systems with promising results.

2.1.5 Virtual Reality (VR)

[28] defined *virtual reality* (VR) as *“a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world)”.* Basically, virtual reality can be considered as a medium, tool and technology, which features some kind of user interaction with a virtual environment. A review of multiple different definitions on VR is provided in the dedicated Chapter 4.2. Related areas like *3D virtual prototyping* respectively *virtual technologies* can be understood as applied virtual reality respectively *virtual reality methods*.

In industrial contexts, the interest in virtual reality focuses on applications which provide particular benefits in the PLM. [29] point that virtual reality has capabilities to reduce time and costs and to increases quality in the development of a product. The authors note that the automotive industry has championed the use of VR across a number of applications, including design, manufacturing, and training. A common example of applied virtual reality in the automotive industry is a digital 3D product model which can be used as a virtual prototype of the real world product. The utilization of such virtual prototypes for different use cases like design review, analysis or simulation can be considered as a typical example of applied virtual reality methods. A more in-depth investigation of virtual reality is proposed in the dedicated Chapter 4.

2.1.6 Augmented Reality (AR)

Augmented reality (AR) superposes respectively enriches the physical, real-world environment with computer-generated elements. For instance, such elements can be graphics, video or sound. Augmented reality is similar to the concept of *mixed reality* (MR) that describes the blending between virtual reality and the real, physical world. Literature like [30] and [31] show that there are many advances in AR research. [32] provide a survey on current state-of-the-art of technology, systems and applications in the field of augmented reality. Also, [33] have reviewed proceedings in AR technology regarding tracking, interaction and visualization. A current and popular example of augmented reality is the smartphone app *Pokémon Go* which shows a successful application in the entertainment industry [34]. Approaches like [35] illustrate benefits of AR for automotive applications, comparing construction data with real cars using visualization schemas. For instance
Figure 6 shows an augmented reality application where the engine bonnet is highlighted in green color.

Figure 6 – Augmented reality example showing virtual data (color variant) being projected onto a physical car front end [36]

2.1.6 Simulated Reality (SR)

Simulated Reality (SR) is a term that emphasizes simulation within the context of virtual prototyping and virtual reality. [37] have defined this term as a new concept for the interplay between simulation, optimization and interactive visualization and, respectively, as a metaphor for the interactive visual exploration of simulation results. According to this definition, simulated reality mostly focuses on enhancing product behavior simulation by methods of visualization and interaction. In their work, the authors have approached the vision of SR by combining design of experiment methods, metamodeling, new interpolation schemes and innovative graphics methods to enable user interaction with simulation parameters and optimization criteria to improve design optimization tasks in automotive crash simulation.

2.1.7 Virtual Technologies (VT)

The term of *virtual technologies* is used in the automotive industry as a synonym for applied virtual reality methods and applications, usually involving 3D visualization, simulation and interaction. These methods are an important resource in the automotive product lifecycle management [38-41]. Virtual technologies help to master the increasing product complexity and they can be beneficial in many aspects like product quality, time-to-market and cost competiveness. In comparison to some scientific definitions on virtual reality, the term of virtual technologies can be considered as a less restrictive description of virtual reality methods that seamlessly blends with many CAx applications. In this thesis, virtual technologies are synonymously used as a comprehensive term that describes the utilization of virtual and augmented reality methods in support of virtual prototyping.

2.1.8 Computer Supported Cooperative Work (CSCW)

The interdisciplinary research field of *computer supported cooperative work* exposes the use of technology in support of human collaboration. "*In its most general form, CSCW examines the opportunities and effects of technological support for humans involved in collaborative group communication and work processes*" [42]. CSCW involves sociology, psychology, computer sciences, media sciences, economics, anthropology, and other thematically related fields of research. In particular, CSCW research addresses social entities (e.g. teams, communities and organizations), social interaction (e.g. cooperation, coordination, coexistence and communication) and utilized tools within this context (e.g. e-mail, video conference systems, groupware, social networks and communication devices). An activity-based definition on CSCW is presented by [43] who defines CSCW as "*computer-assisted coordinated activity such as communication and problem solving carried out by a group of collaborating individuals*". [44] propose that "*CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work*". [45] noted that there can be divergence in the understanding of definitions, contents and targets of CSCW research which is likely to be valid still today.

2.1.9 Computer-mediated Communication (CMC)

In many industrial branches, team-oriented collaboration and discussion are integral parts of daily work. Such collaboration and communication tasks are increasingly supported by *computer-mediated communication*. Many different tools like displays, conference systems, groupware, and hand-held devices are utilized to aid in communication. Moreover, in the manufacturing industries such as the automotive industry, particular technologies and devices are required to appropriately visualize and communicate data related to virtual product prototypes. However, while there is a large array of devices, there is still little understanding of how and when to use them most effectively [46]. Moreover, new media raise the question which medium is appropriate for which type of collaboration [47]. This question is particularly important for industrial application which aims to identify most appropriate tools and technologies for efficiently supporting communication as well as streamlining daily processes and thus improving profitability.

2.2 Data Structures

In CAD-based automotive development, many different systems and file formats are utilized along product lifecycle management. This section explores a set of data structures which are relevant in the context of this thesis.

2.2.1 CAD Data

In this thesis, the term *CAD data* is particularly understood as a set of data that includes a geometric 3D representation of a virtual product prototype or a part of it, and optionally including a set of product properties. For example, Figure 7 shows geometric CAD data of a vehicle exterior.

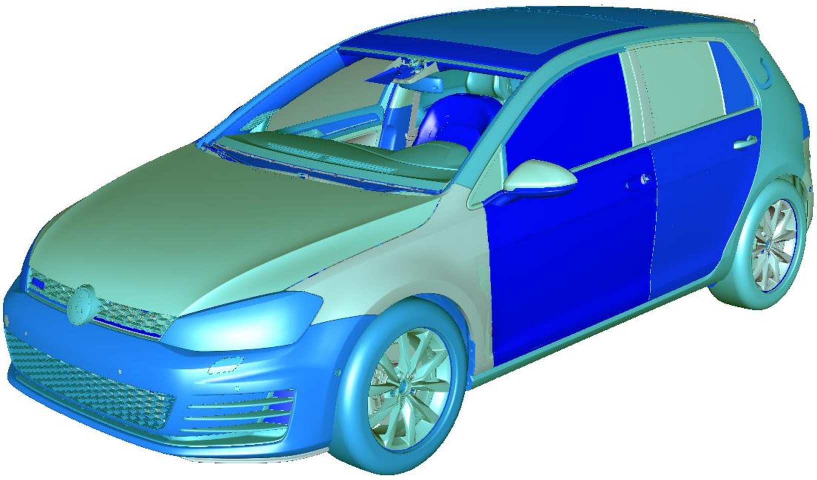

Figure 7 – 3D CAD data of a vehicle's exterior (a virtual product prototype).

For another example, Figure 8 shows CAD data of a single vehicle part (brake disc).

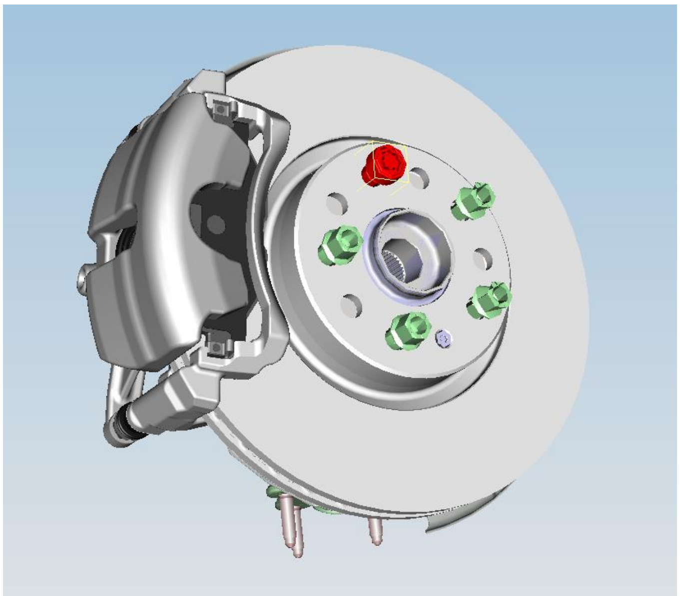

Figure 8 – 3D CAD data of a vehicle part (brake disc)

Process-wise, CAD data is originally created by engineers in a CAD system and afterwards usually stored in a *data management system* (DMS) to make the data available for other stakeholders and further applications like digital mock-up and manufacturing. Generally, in automotive product lifecycle management, data management systems assist in tasks like information enquiries, storage and provision of data, version and configuration management and data exchange with internal and external business partners.

The CAD data used in this thesis is stored in an established data management system in the field of automotive product lifecycle management. This system stores the CAD data in predefined sets whose content is related to different scopes such as vehicle projects, assembly zones, car configurations and other extents. Such volumes can include car segments like front end, cockpit, complete car configurations, and configurations which include multiple variants and derivatives. For example, Figure 9 shows a few examples of geometric CAD data which is related to different assembly zones.

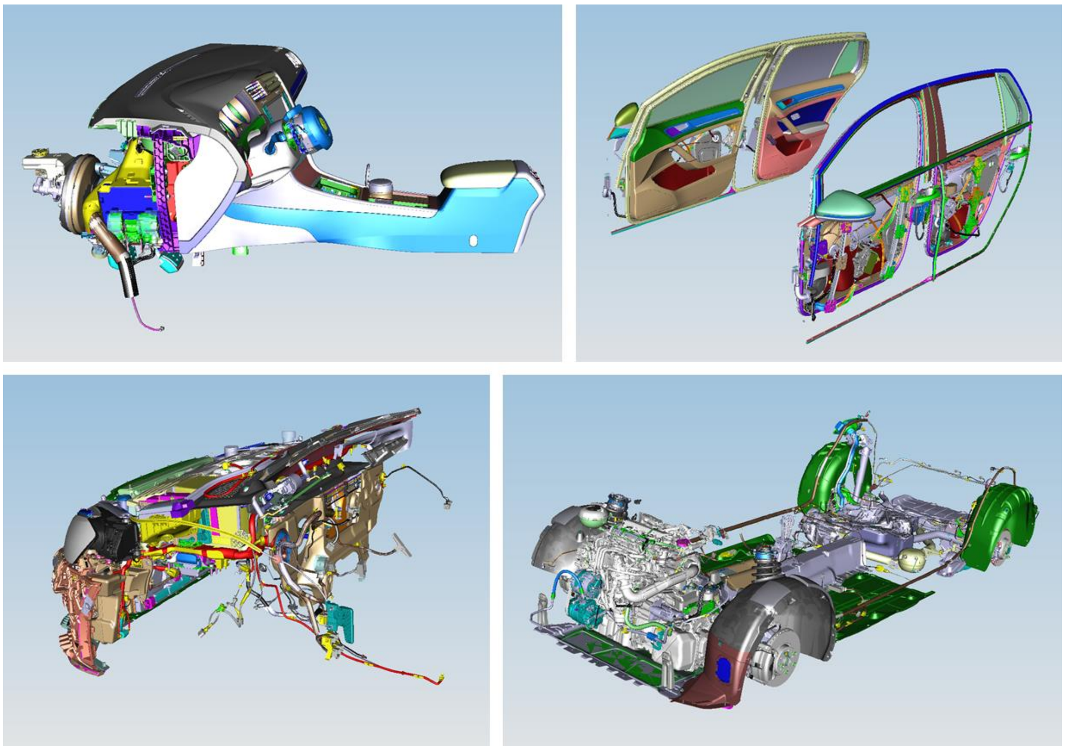

Figure 9 – Examples of CAD data volumes based on four different vehicle assembly zones

Notably, a complete CAD model would represent a specific car configuration including all parts which are required to physically assemble the vehicle in the real world. Moreover, a CAD model also could include all possible variants and configurations and thus include more components than could possibly be assembled within a single vehicle in the real world.

For the research work of this thesis, a well-established 3D visualization system has been used for utilization and visualization of 3D CAD data as shown in Figure 7, Figure 8 and Figure 9 [48]. This system is used for two reasons. First, the system is used by the OEM that has supported this thesis work and it is already implemented in current development processes and DMU applications. Thus, many users are familiar with this system and thesis results can be easily integrated into existing workflows. In addition, it offers a comprehensive and sufficient amount of features for 3D visualization and geometric data analysis so that it provides an appropriate technical basis for further methodical research.

A typical 3D CAD-based visualization system usually provides methods to identify data structures of the geometric data. In particular, the visualization system used in this thesis provides a hierarchy tree window which shows the structural arrangement of the geometric data as shown in Figure 10.

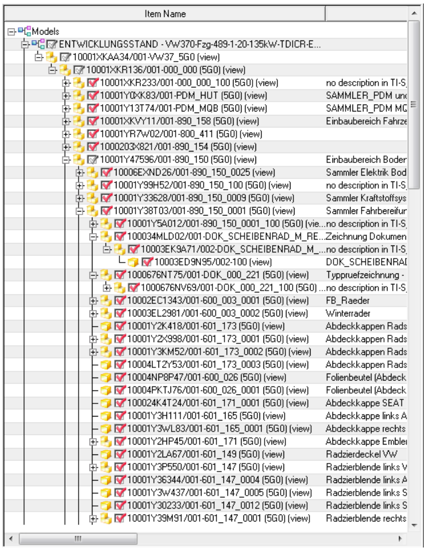

Figure 10 – Example of a hierarchy tree that represents a CAD data structure.

The smallest differentiable piece of geometry within the utilized CAD data record is called a *part*. Parts may contain additional information. In this example, such information can be accessed in the visualization system via a properties dialog. Some of these properties are automatically set by the tool, some can be manually set by the responsible CAD engineers. Notably, the information available in the properties do not include any function-oriented information. For example, for a given part, it is not possible to get information about the mechatronic type of the part (e.g. sensor, actuator, controller) or the vehicle functions where this part is used in.

In this thesis, the term of *granularity* is used to describe the degree of segmentation of the geometric parts within a set of CAD data. A low granularity means that multiple parts of geometric data are merged into one single part as shown in Figure 11 (left, blue wiring harness). Especially wires are frequently merged to simplify DMU applications because such applications usually focus on geometric analyses and therefore do not require a high granularity. In contrast, the right side of Figure 11 shows a high granularity, involving a fine segmentation that separates different wire segments as individual geometric parts (see colored wiring harness). In this case, different segments of the wiring geometry can be individually selected and utilized.

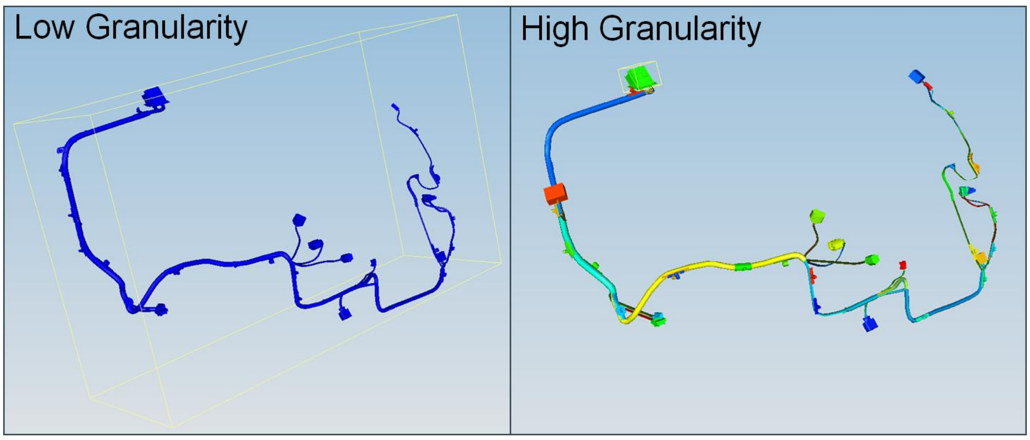

Figure 11 – Comparison of different granularity of CAD data elements. Low granularity (left): geometry is a single part, high granularity (right): wire segments are separate parts

2.2.2 Jupiter Tessilation (JT)

Jupiter Tessilation (JT) is an established industry-driven 3D CAD data format, used for product visualization, collaboration and data exchange. The JT format is a common and widely accepted industry standard defined in ISO 14306:2017 which provides a high level of interoperability and compatibility [49].

The JT format is a product of the JT Open Initiative which is a community of independent software-providers that agreed to the use of an open and JT-based 3D visualization platform to channel the requirements of different domains [50]. A primary aim of the initiative is the technical collaboration of individual companies along the PLM with particular focus on visualization, collaboration and standardized interoperability. The resulting JT Open Toolkit provides a free C++ library which enables any CAD-applications to read, create and process JT-files. This toolkit is free to use for members of the JT Open Initiative. Further read is to be found online [50].

Technically, JT is a platform and tool-independent 3D data format capable of representing a wide range of geometric engineering data. The format uses a scene graph with CAD-specific nodes and attribute-support. This data can be very lightweight, holding little more than approximate facet data, or, it can be quite rich, containing complete boundary representation surfaces (NURBS), geometry representations along with product structure, attributes, metadata and manufacturing information. It also supports multiple tessellations and level-of-detail (LOD) generation. JT data can be structured in assembly groups with multiple hierarchy levels and additional information like product properties can be integrated. A container structure enables engineers to edit single parts without the need of loading the complete virtual product. Limitations of the JT format include a loss of the construction history and file version. Free examples of JT and other CAD files are available on websites such as *GrabCAD* and others [51].

2.2.3 XML

XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The format is defined in the XML 1.0 Specification provided by the World Wide Web Consortium [52]. Advantages in its use for data are simple syntax, support for nesting, easy to debug and language- and platform independence. The XML format is universal, transferable via web and intranet and widely accepted and distributed. According to [53], XML is able to support interactive data exchange between different sources and it’s characteristics like self-definition and extensibility make it capable of expressing almost all kinds of data. [54] point out that the most common applications of XML today involve the storage and transmission of information for use by different software applications and systems. The XML format can be an appropriate solution if an interoperable and system-independent utilization of data is required.

2.2.4 PLM XML

The *PLM XML* file format has been developed with the aim to facilitate the interoperability in the product life cycle by the use of XML (based on the WC3 XML standards) [48]. By supporting both, direct and referenced mapping of product data, PLM XML provides a compact, expandable and flexible file format for the description of product information. PLM XML is frequently used as a complementing format to JT in terms of defining product structures, custom data and process information in a human-readable notation. In contrast to JT, an advantage of PLM XML is that the format is easily editable and readable without the need for custom tools, APIs and/or interfaces.

The PLM XML format is particularly relevant for the work of this thesis because it provides an efficient solution for a streamlined integration of individual product hierarchies and custom metadata with CAD data. In addition, PLM XML is an integral part of the Siemens PLM software which is an established industry standard used by multiple automotive OEMs. Thus, approaches using the PLM XML format are likely to be more seamlessly integrated into existing PLM structures and processes.

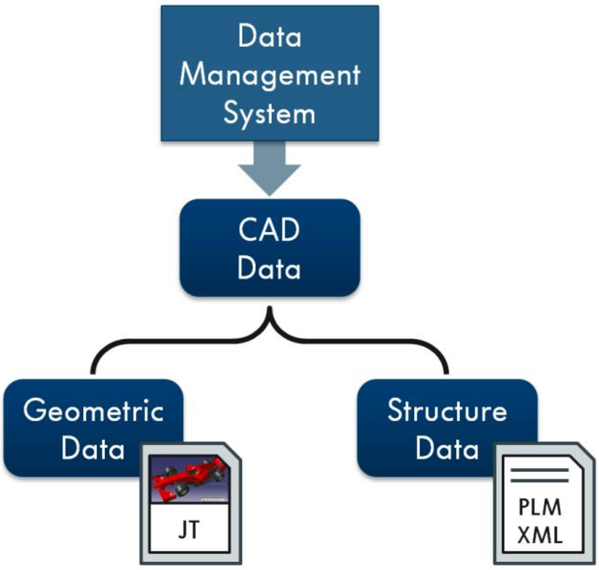

Figure 12 – Combining JT and PLM XML provides a dual data structure which can be beneficial for encapsulating geometric data and other relevant product data

The file formats of JT and PLM XML together provide a dual data structure that enables a separation of geometric data and custom additional data as shown in Figure 12. Moreover, due to its' XML-architecture, the metadata part fulfills the criteria of easy access, human-readability and being editable without custom software toolkits.

3 Introduction to Function-oriented Development and related Challenges

This chapter discusses the recent challenges of an ever-increasing complexity in the automotive industry and, in this context, the approach of function-oriented development is introduced. In addition, proxies of current vehicle functions are described and complexity, diversity, and challenges of a function-oriented development are illustrated.

3.1 The Complexity Challenge in the Automotive Industry

In recent decades, the automotive industry has proven to be one of the strongest drivers of technology, growth and employment [55]. Automotive engineering is one of the key industries in virtually all developed economies and one of the driving forces behind globalization [56]. Nevertheless, today the industry has to face a considerably amount of challenges. For example, global competition has significantly shortened product and development life cycles as shown in Figure 13. [57] noted that vehicles have advanced from being predominantly mechanical systems to increasingly electronic systems. Above all, one of the primary challenges in the automotive industry is the increasing complexity of modern vehicles [1,2].

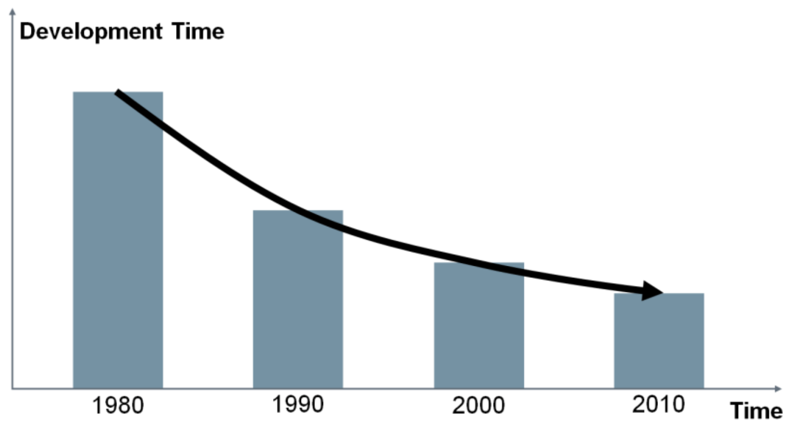

Figure 13 – Reduction of automotive development times in the last decades [24]

A significant driver of automotive complexity is the high amount of vehicle electronics respectively vehicle functions such as *park assist*, *dynamic light assist* or *start-stop automatic*. The amount of different vehicle functions in one vehicle can easily exceed a hundred different functions, involving many areas like safety, drivers' assistance, comfort, infotainment, entertainment and others. For example, in a modern car, the number of functions related to the lights can already include twenty and more functions, ranging from such functions as *Instrument Lighting*, *Parking Light* and *Headlamp Flasher* to *Dynamic Light Assist*.

The past four decades have witnessed an exponential increase in the number and sophistication of functions respectively electronic systems in vehicles as shown in Figure 14. Noteworthy, the cost of electronics in luxury vehicles can amount to more than twenty-three percent of the total manufacturing cost [58].

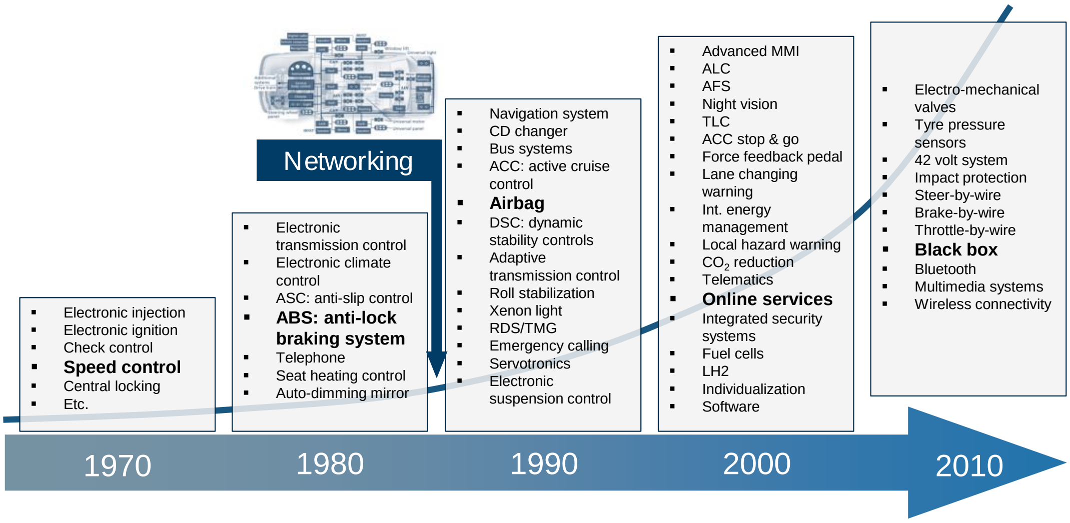

Figure 14 – Increase of vehicle functions in the last four decades, based on [57] and [58]

[59] underline that the growth of system complexity in the automotive industry is mainly driven by the fact that product functions are more and more implemented by the combination of mechanical, electric/electronic and software components. Technically speaking, many modern vehicle functions are implemented as *mechatronic systems* consisting of sensors, actuators and controllers, respectively *electronic control units* (ECUs) [60]. The *Verband deutscher Ingenieure* (VDI)) defines mechatronics as *"the synergetic integration of mechanical engineering with electronic and intelligent computer control in the design and manufacturing of industrial products and processes. Mechatronic systems comprise a basic system, controllers, sensors, actuators and information processing. Also of significance is the environment in which the mechatronic system is operated."* [61].

Today, new generations of embedded systems in modern vehicles are increasingly considered as *cyber-physical systems* (CPS). Cyber-physical systems involve transdisciplinary approaches, combining cybernetics, mechatronic design, and design and process science [62-64]. In contrast to traditional embedded systems, cyber-physical systems put a focus on connectivity and networks of interactive elements with physical input and output instead of as standalone devices [65]. A common example of a CPS is a sensor-based communication-enabled autonomous system with connectivity to other systems.

Another significant complexity driver in the automotive industry is the high amount of different variants due to different technical systems (e.g. engines, transmissions, steering controls), markets and the possible configurations including multiple customizable options. For instance, in Germany, typical mid-range cars are currently available with options for different engines, transmissions, assistance systems and some provide options for more than 100 optional features, resulting in a total of multiple thousand possible configuration variants for such cars only for the German market. [24] underlined that today cars are no longer mass products featuring uniform color, shape and equipment. Moreover, companies can no longer follow the trail blazed by Henry Ford, capturing market share and high profits by producing large volumes of a standardized product [66]. Instead, since multiple years many companies have stopped to use a mass production model and have adopted new, flexible principles for organizing work that have demonstrable advantages in terms of economic performance [67]. Consumer preferences determine the current styles, reliability, and performance standards of vehicles, resulting in an increasing amount of projects, modules and variants. Consequently, this high diversity of product variants necessarily leads to an increase in complexity of all operational structures and processes [22]. In addition, legal conditions and constraints have tightened, and customers are constantly becoming both, more demanding and more varied in their individual preferences [68].

In terms of technical design, it is noteworthy that actual technical implementations of vehicle functions as mechatronic systems can be considerably different in their system architectures, software usage and involved components. Such technical variations do not only depend on vehicle models but also on particular derivatives and configurations within the same vehicle models. For example, a simple function like *seat heating* can be considered. Depending on the vehicle configuration, this function's software could be implemented in a dedicated ECU, or within a central ECU which holds many other functions (e.g. body control module). As a result, the high diversity of different variants and implementations additionally increases electronic complexity and thus provides considerable challenges for automotive development, testing, quality assurance, service and other domains. For example, a service technician who needs to identify and fix a vehicle defect might be confronted with a different technical system implementation for each particular variant so that the repair task can become more difficult and requires more knowledge and experience as it can be different for each particular variant.

A current trend is that the number of electronic systems in modern vehicles is continuously growing while at the same time systems are getting more complex. While the first ECUs in the eighties were only required to work properly within their own borders, the introduction of inter-vehicle networks introduced the challenge that ECUs were additionally required to work properly in mutual dependency within the inter-vehicle network. In current vehicles, the number of ECUs can vary from about 50 to 80 ECUs, depending on the vehicle model and configuration. Embedded software in ECUs continues to increase in line count, complexity, and sophistication [69]. For example, the share of software working in each ECU can involve up to hundreds of thousands of lines of code per ECU which all need to work properly in order to guarantee appropriate vehicle functionality and assurance of functional safety. In total, the share of software of modern vehicles can exceed 100 million lines of code [70].

A new layer of complexity has been recently added because electronic communication of modern cars is not even limited anymore to inter-vehicle networks but also involves external communication like the internet, mobile phones, other cars and various other entities. Notably, electronic communication with external entities always means data exchange with a potentially unknown and unpredictable environment. Thus, the proper implementation of such external system interfaces provides a significant challenge for the industry in terms of both functional safety and car security. It can be argued that an insecure car, which for example can be hacked, is an unsafe car as well. Such issues pose major challenges in terms of functional safety and security so that proper verification, validation and quality assurance of vehicle systems becomes more important than ever before. Therefore, today, the need for appropriate development processes and holistic validation strategies is further increasing.

Summarizing, among other things, automotive product development is today affected by global competition, short product life cycles, increasing demands for quality, safety, security, innovation, environmental and political defaults. Most of all, there is a high and further increasing product and development complexity caused by the increasing amount of vehicle functions, electronics, software usage, configurations and networking. Thus, managing the increasing vehicle complexity has become a key challenge for both *original equipment manufacturers* (OEMs) and suppliers.

3.2 Defining Function-oriented Development

A typical OEM definition of a *function* in an automotive context can be as follows:

"A function is a vehicle behavior/feature that is recognizable for a vehicle passenger, road user or service employee and which provides a concrete benefit for the customer from a marketing point of view."

In an automotive context, *function-orientation* can be considered from different perspectives. From a marketing perspective, the function-oriented approach closely takes into account a customer's point of view. Customers rather consider vehicles as products with specific features and functions instead of focusing on particular components. For example, functions like an innovative infotainment system or a safety-increasing driver assistance feature can be important factors to increase attractiveness of a vehicle product. Thus, the range of customer-visible vehicle-functions has a major impact on the competitiveness of the product.

From a design perspective, a function-oriented approach of development seeks for an efficient way to develop and implement vehicles which involve a significant number of different functions. In this context, the term of a *function* can be initially considered from a view that is still uncoupled from specific vehicle projects and technical implementations. So, from the design perspective, a function can be specified by describing a product behavior on a high-level layer without the need to know which particular vehicle projects will use it, or how it is particularly implemented. For example, a park assist function would be usable in any kind of usual passenger vehicle and it can be specified and described without the need to go into the details of a vehicle-specific technical system realization.

In contrast to a function, a *system* is understood as the actual technical implementation of one or more vehicle functions with all of its related components. In this context, components can be actuators, controllers/ECUs (synonymous), sensors and mechanical parts. Single components can also be part of multiple systems at the same time. Notably, the meaning of the term *system* can be different, depending on the context. For example, the above definition is the function-oriented understanding of a system as a group of components which implements a function. However, the overall vehicle can be considered as a highly sophisticated system as well. Also, a particular component like an ECU can be considered as a complex system, involving its own sub-systems and sub-elements like electronics, mechanics and software.

Figure 15 illustrates the dependencies and differentiation between the *function*, *system* and *component* layers in the function-oriented context.

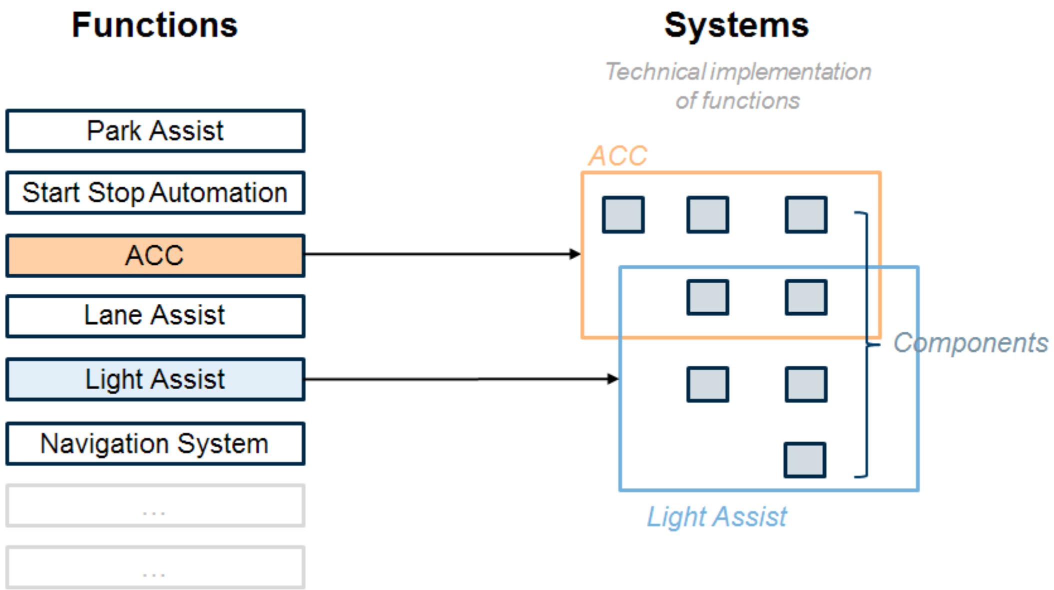

Figure 15 – Vehicle functions (features) are implemented as technical systems, consisting of multiple components. Particular components can be part of multiple systems. Components are systems as well, consisting of mechanics, electronics and software

Systems, respectively technical implementations of vehicle functions, can be considerably different, depending on particular vehicle projects, configurations and variants. This principle is illustrated in Figure 16. As described in Section 3.1, this kind of diversity increases complexity for many automotive domains, especially for development, quality assurance, production, distribution and customer service.

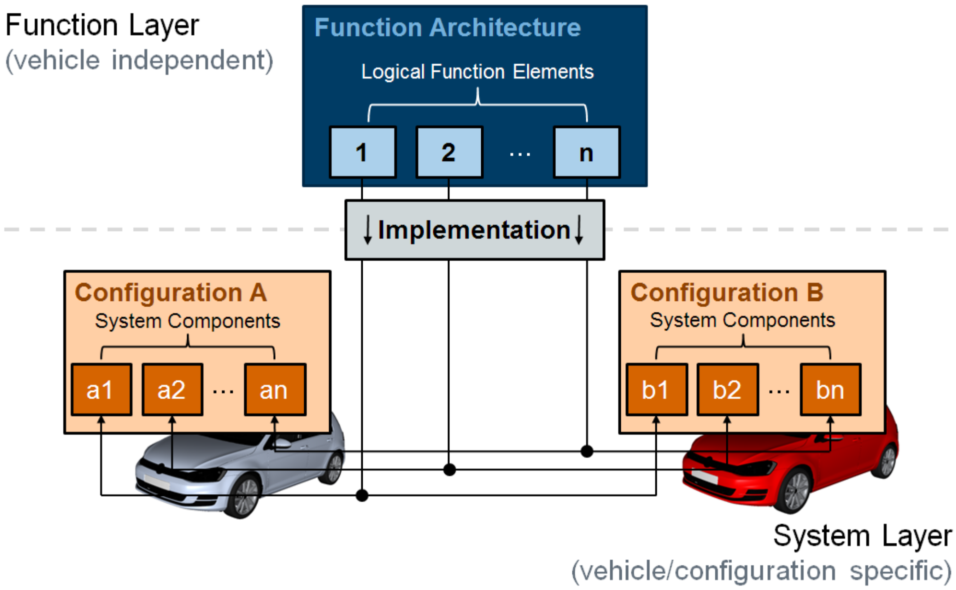

Figure 16 – Systems, which are technical implementations of functions, can be different among different vehicle configurations, variants and derivatives

Many vehicle functions are realized as interdisciplinary projects, involving multiple different development departments and domains. For example, a start-stop automation system requires interaction of such vehicle components like engine, transmission, brake and steering wheel. In that context, the need for coordination and communication among participating departments of development rises with the complexity of mechatronic systems [71]. Thus, there is a strong need for appropriate development processes which are capable of handling the high degree of complexity.

For a long time, the automotive industry exclusively followed a component-driven approach of development. Today, however, it is understood that the development of distributed and highly complex electronic and mechatronic automotive systems cannot be reliably handled with a purely component-driven development any more [72]. A major reason is that component-driven approaches of development focus on single components rather than on overall integrated systems and functions. However, the validation of single components does not necessarily assure their proper interaction altogether as parts of comprehensive, distributed and highly complex systems. Moreover, many aspects like mutual dependencies, synergies and reusing potentials are not visible from a purely component-oriented perspective. Thus, additional layers of abstraction are required in development processes which address those layers above the component level.

A promising solution is the relatively new *function-oriented* development approach. The function-oriented approach extends the overall development focus on distributed vehicle functions and focuses an interdisciplinary implementation of mechatronic and cyber-physical systems. The function-oriented approach aims at mastering complexity with the principle of modularization and clearly defined processes and abstraction layers within development and validation. It is a top-down approach where the functions and systems specify the needed components as shown in Figure 17.

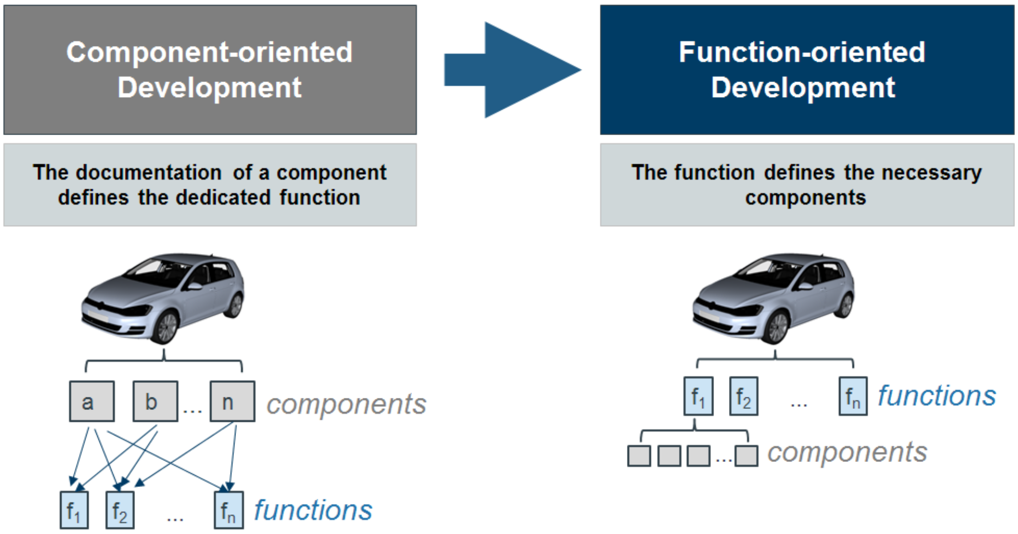

Figure 17 – From component-oriented to function-oriented development

Automotive function-oriented development processes involve multiple abstraction layers including vehicle, functions, systems, and components which form the systems. The processes required for the implementation of vehicle functions using the above layers can be modeled with the V-model. The V-model is a comprehensive, abstract methodology for project management that is originally known from the field of software engineering. It provides a method for the accomplishment of complex projects. The left branch of the model deals with specification, including requirements specification, architecture design and implementation, while the right branch focusses on verification and validation, respectively on testing. Notably, the V-model is an iterative model which means that full branches or parts of it are usually passed through multiple times. For example, if a system requirement is changed, it can have an impact on dependent work products like architecture, software implementation and test cases so that all related process steps need to be reiterated after the change. An official documentation of the V-model is available online [73]. Figure 18 shows a high-level adaption of the V-model I have created for this thesis to illustrate the function-oriented development process.

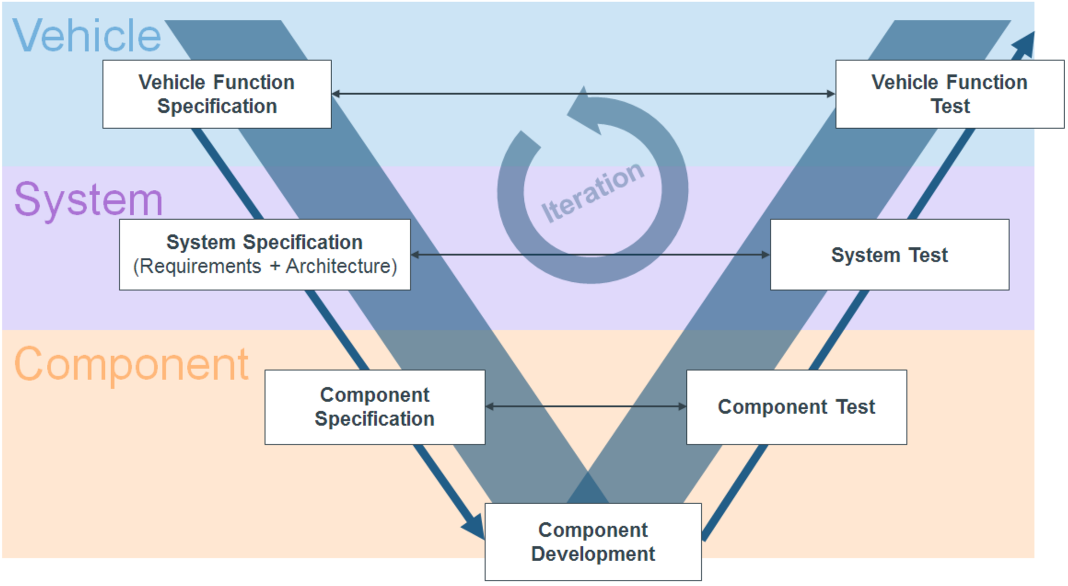

Figure 18 – An adoption of the V-model for function-oriented development, illustrating vehicle, system and component layers. The left branch deals with specification, the right branch deals with verification and validation.

In order to illustrate a function-oriented development process based on the V-model, for example, it is assumed that a new vehicle project is started. So, on the vehicle level, it needs to be specified which functions are to be featured within this specific vehicle. This selection considers aspects such as strategic, market-related and economic decisions, customer and market requirements and safety, legal and economic constraints. Particular functions can be chosen and assigned from those which have already been defined (e.g. for prior vehicles). In addition, novel functions can be defined for the new vehicle project. As a result, a feature specification is created which defines all the functions which are supposed to be implemented in the new vehicle. For each function, a high level function description needs to be available and dependencies and interfaces between functions must be identified.

On the system level, specifications need to be created which describe the systems which implement the featured functions, taking into account the individual constraints and assembly situation of the particular vehicle project as well as its variants and configurations. The system specifications need to include all system requirements which describe the intended system behaviors, including functional and non-functional requirements. In addition, system architectures need to be specified which define sub-systems, involved components, interfaces and dependencies between all system elements. In addition, high-level functions need to be divided into particular sub-functions and they need to be assigned to one ore multiple ECUs by the related system architectures.

If the system designs are finished, system-related components either need to be newly developed or chosen from those which are already existent. Reuse of components has multiple significant advantages such as saving of development costs and time, saving of assembly spaces and reduction of vehicle weight. For example, it is common practice that sensors and actuators are used by multiple functions at the same time. Moreover, chances are that new functions can be implemented without the need to add new components at all but by just making changes to the software of an already-existing ECU. Therefore, a significant effort is put into modularization to enable synergy effects and to allow reutilization of already existing and approved components.

On the component level, each required component needs to be specified with a clearly defined requirements specification. Usually, most ECUs are not developed by automotive OEMs but by suppliers which are specialized on development of automotive ECUs. Notably, development of ECUs is usually based on the V-model as well, including respective processes for specification, implementation and testing. Each particular ECU in a vehicle consists of hardware, electronics and software and it usually is a highly complex system on its own.

Dedicated processes for validation and verification need to be performed for each particular layer within the V-model. So, all components, systems and vehicle functions need to be verified with appropriate verification methods in order to ensure that they fulfill the respective requirements and that they can be properly integrated into the vehicle.

The function-oriented approach strongly uses the principle of modularization across all abstraction layers. The need for modularization is based on the fact that systems with too high complexity cannot be handled and mastered by humans any more without proper methods. So, the goal of modularization is to sub-divide complex systems into smaller modules until they have a size and complexity that can be reasonably mastered. This principle is illustrated in Figure 19.

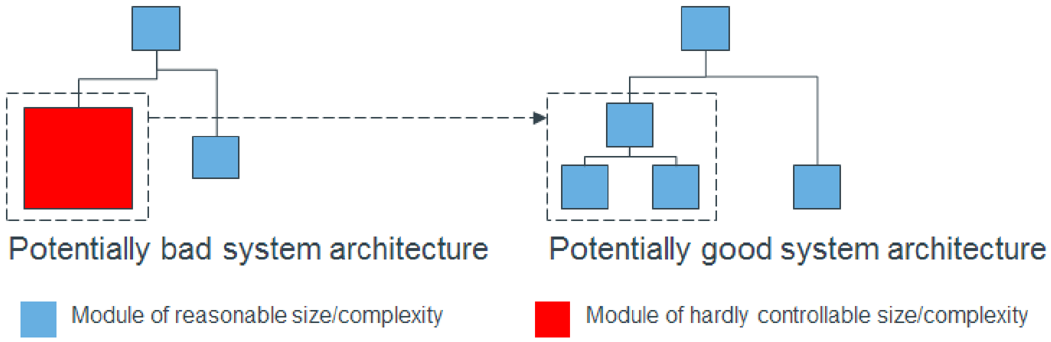

Figure 19 – The principle of modularization helps to reduce complexity by dividing complex systems into multiple smaller modules of lower complexity

Modularization also allows for more efficient testing strategies, including individual verification and validation of particular modules. If particular module tests are passed, such modules can be integrated incrementally and tested again by integration tests which primarily focus on validating module interfaces based on architectural specifications. So, different test layers allow to identify different types of issues at preferably early times and to ensure requirements fulfillment and consistency between specification and system implementation. In addition, in case of later changes or problem resolution measures, modularization helps to improve such processes. For example, if an issue is detected, at best it might be sufficient to only analyze, revise and/or exchange a single module. Due to the high amount of software in current automotive ECUs, it is a key process to design appropriate software architectures.

Summarizing, the function-oriented approach is a particular form of systems engineering. Considering the state of the art, this and comparable development systematics provide the most fundamental measures to master the high and further increasing automotive complexity by defining appropriate and necessary processes for development of highly sophisticated mechatronic cyber-physical systems. This top-down approach uses clearly defined processes, abstraction layers and modularization in order to identify synergies, dependencies and reuse potentials and provides the capabilities to improve testability and quality of such systems.

3.3 Typical Vehicle Function Examples

In this section, a selection of typical vehicle functions is described. Some of them are used as examples throughout the thesis and also in the contributions in the later chapters. These functions are selected with regard of being representative, understandable and different in complexity and area of application.

3.3.1 Headlamp Flasher

This function is chosen to provide an example of a simple, well-known vehicle function that most people should be familiar with. To trigger this function, the driver usually needs to push a button somewhere near the steering wheel. The function then temporally increases the brightness of the headlights and/or flashes additional vehicle lights. This function is supposed to be used as a warning and notification light, or as an alternative to the honk.

3.3.2 Adaptive Curve Light

An adaptive curve light system improves the drivers' visibility by the automatic control of light distribution to suit environmental conditions. Such systems are supposed to create an optimal illumination of the drivers' field of view in any road situations. For example, the system might automatically pan the front lights when the driver enters a curve with a radius smaller than approximately 300 meters and when the cars' speed is higher than a specific minimum speed. The panning distance is calculated depending on the current steering angle. The system usually is only active at darkness and when the lower beams are enabled. Figure 20 illustrates the basic principle of such systems:

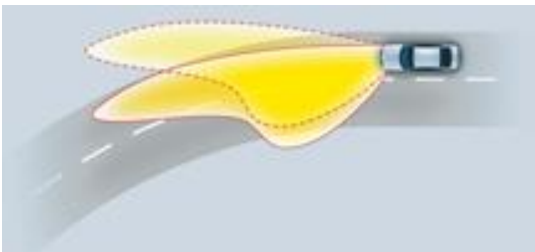

Figure 20 – Adaptive curve light is a modern vehicle function which improves road illumination in curves as illustrated in this graphic [74]

3.3.3 Parking Steering Assistance

Parking steering assistance systems, shortly called park assist in this thesis, scan the space of a parking lot and automatically maneuver the vehicle in. Usually, ultrasonic sensors in the front bumpers scan the parking space to detect whether it is big enough. Then, for example, the automatic, parallel park maneuver can be activated by pressing a single button. The driver stops the car, selects the reverse gear and let go the steering wheel. The park assist function shows the driver the intended reverse path on the multifunctional display and puts the car into an optimal starting position. Then, it automatically steers the car into the space. Though the driver does not need to steer he still stays in control of the car by working the accelerator or brake and clutch. The system can be immediately deactivated by taking over the steering or braking to a standstill (see Figure 21).

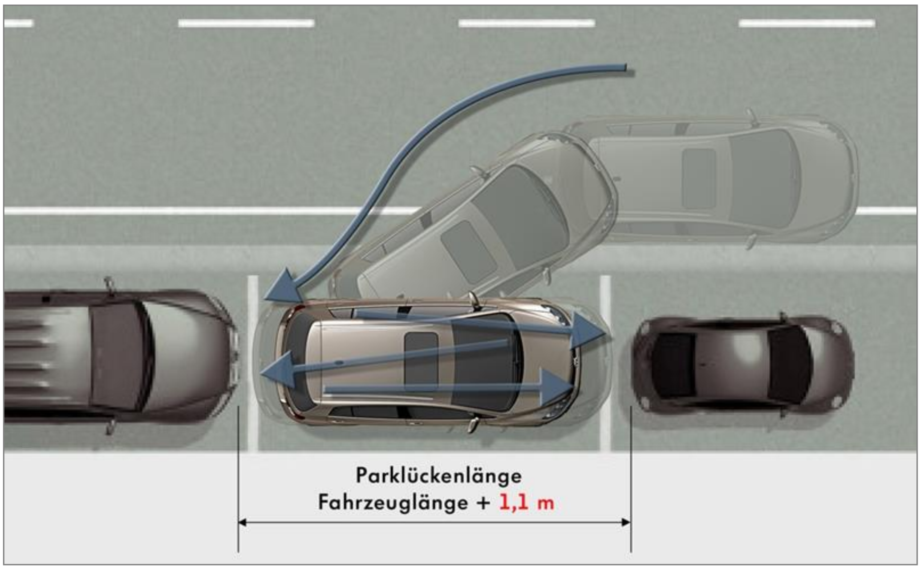

Figure 21 – Parking steering assistance is a vehicle function that assists the driver with automatic steering into a parking lot [75]

3.3.4 Direction Indicator (Turning Light System)

Direction indicator functions are primarily used to indicate drivers' intents for vehicle turning maneuvers. While respective systems can be implemented differently across automotive OEMs, such functions are usually part of superior light signal system that might also be used for different other tasks, including but not limited to light-based indications for emergency situations, anti-theft alarm and central-locking systems. A turn-signal light system could be used in multiple prioritized functions with a function like hazard flashing possibly being of the highest priority. For example, a typical hazard flashing functionality would cause all turn-signal lights to flash synchronously. In case of light failures, a defect lamp might cause a doubling of the flash frequency and/or would be indicated as a failure in the instrument cluster display. Technically, the turn-signal lights could either be directly operated by a central ECU or by other ECUs which are responsible for particular other vehicles systems such as door or trailer control and which require some of the turn signal systems lighting functionalities.

3.3.5 Start-Stop System

A start-stop system is an economic function that helps to reduce the fuel consumption of cars by automatically turning off the engine in standing situations like in front of red traffic lights. Usually, the engine is turned back on when the driver wants to move on. Technically, start-stop systems are usually rather complex because they may involve a considerably high amount of related components and need to interact with multiple other vehicle systems. For example, a particular implementation of a real world start-stop system includes about fifteen controllers, including electronics for engine, transmission, brake and steering wheel.

3.4 Introduction to Function Architecture Design in Automotive PLM

Architecture design is an important step in the technical development of automotive functions, respectively in the development of the mechatronic systems actually implementing these functions. Within this thesis, the term *function architecture* denotes a methodology that allows to identify all components that make up a specific vehicle function, respectively a system that implements this function, including the connections between these components. Within these architectures, components can be sensors, actuators and controllers/ECUs. Connections between components may include but are not limited to controller area network (CAN) busses, signal wires, ground cables and supply lines. So, a function architecture should provide a means to identify all components and electrical connections in the vehicle necessary to implement a particular function, for instance, the wipers (a simple example with about 10 components) or the park assist function (a rather complex example with more than 30 components). In this thesis, function architectures are also shortly called *function architecture data*.

In automotive electric/electronic development, there are many aspects that are concerned in decisions regarding architecture design. For example, such aspects can be distribution of the function in the vehicle, choices and placement of components, usage of communication systems (buses), power concepts, segmentation of wiring harness, timing issues, cost factors, weight factors, integration of subsystems into higher level vehicle architecture and optimization issues. The priority of such aspects can be dependent on the particular requirements of the function and the vehicle. Technically, different software tools are available to support the process of function architecture design including different layers of abstraction to subdivide information and to fit respective use cases.

In this thesis, a focus is set on the information provided by *architecture diagrams*. In the design of function architectures, architecture diagrams are used to describe and specify automotive systems including their constituting components and connections. Figure 22 illustrates a slightly simplified architecture diagram of a vehicle *headlamp flasher* function. The diagram shows an architecture involving different types of components and connections, including controllers (blue), sensors (purple) and actuators (orange).

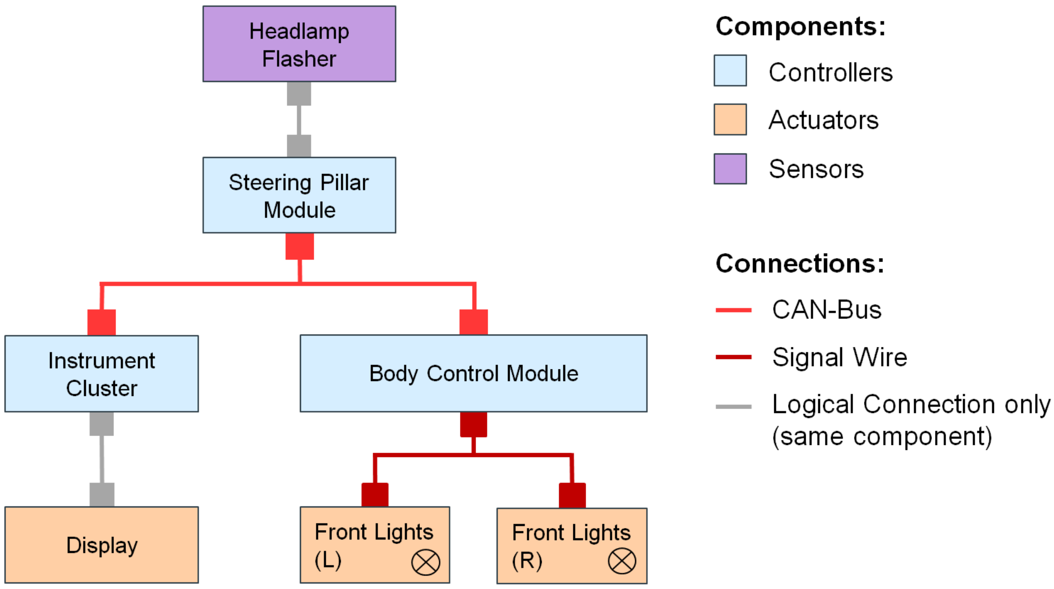

Figure 22 – The function architecture diagram of an automotive headlamp flasher function. The thesis approach aims at enabling a spatial visualization of such function architectures in actual CAD-based vehicle configurations

In this headlamp flasher architecture, the function is triggered by a headlamp flasher switch (sensor) attached to the steering column module (ECU). The signal is then communicated over a CAN bus to other controllers, forwarding the signal to the particular actuators which in this case are the front lights and a control lamp in the instrument cluster display.

As noted before, actual system implementations of vehicle functions can be different, depending on vehicle models as well as on particular vehicle configurations and variants. For instance, a specific function could be implemented in a central body control module (BCM) which is already responsible for other tasks, or within a dedicated and encapsulated ECU. The architectural choices of distributing functionality over components can depend on multiple factors such as costs, system robustness, configurations, availabilities, software dependencies and many more. Thus, the very same function could also be realized with a different architecture. For example, the functionalities of the three ECU's could also be combined in a single ECU.

For the specification of architecture variants, either different architecture diagrams can be created or single diagrams can involve architecture alternatives. For example, Figure 23 shows a headlamp flasher system similar to Figure 22. In contrast, however, this version also includes variants for the front light actuators which could either use standard or Bi-Xenon lights. The variants are separated with logical XORs.

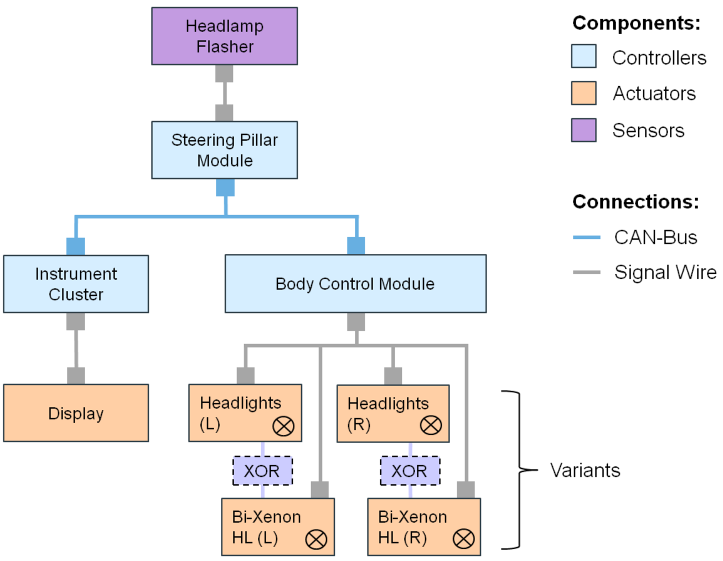

Figure 23 – Hypothetical headlamp flasher function architecture with variants

In this section the headlamp flasher function has been used as an example to discuss function-oriented architecture design. Notably, the headlamp flasher is a comparatively simple function and other functions can be of significantly higher complexity. For example, Figure 24 shows an exemplary architecture of a park assist function that includes a considerably higher number of components. As it has been discussed in Chapter 3.1, due to the current increases in complexity of cyber-physical systems it is likely that function architectures will further increase in complexity. Notably, there are approaches that aim to reduce the overall number of vehicle ECU's while maintaining the same functionality. However, this does not necessarily mean a reduction of complexity because complexity is mostly shifted across different technical layers such as system and software architecture.

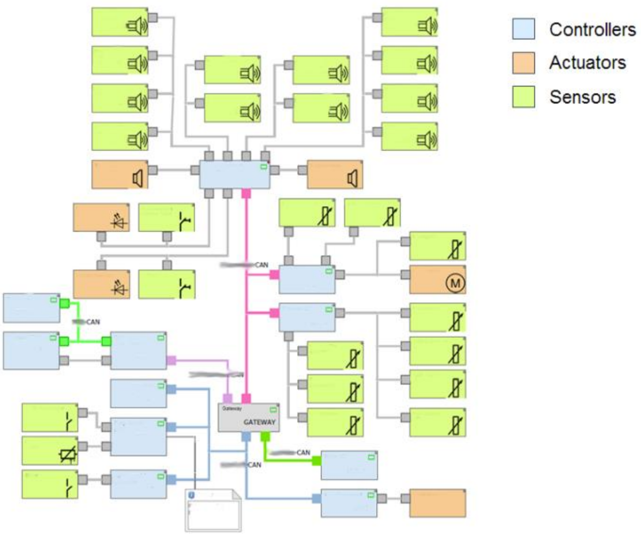

Figure 24 – Exemplary function architecture of an automotive park assist function which features a considerable architectural complexity

Notably, function architecture diagrams are not only used in development but also in multiple other areas of the automotive PLM including research, validation, quality assurance, production, maintenance, customer service and the legal system. Thus, they make up an important part of vehicles system specifications and provide a key work product that is used in multiple relevant tasks within function-oriented development.

3.5 Geometric Function Localization Task (GFLT) Definition and current Challenges

The spatial localization of the distribution of components and wiring harness of a specific vehicle's function of a car model is an important task throughout the full automotive product lifecycle. In this thesis, this task is defined as *Geometric Function Localization Task (GFLT).* The GLFT is especially important for a function-oriented development which requires a highly interdisciplinary collaboration between different domains and departments of automotive development such as architecture design, wiring harness development and virtual prototyping. Moreover, since this task occurs numerous times, it is important that users like engineers, designers, testers and service technicians are able to perform it time-efficient and accurate. In addition, due to the extensive range of cross-domain users, it is of advantage if this task requires as little additional expert knowledge as possible.

The GFLT can be crucial in many use cases. Such use cases include, for instance, the design of inter-vehicle networks, evaluation of crucial function aspects, identification of synergy and savings potentials, making statements on functional dependencies on crash zones, communication and the creation of mutual understanding of function architectures, service and maintenance, and the function-oriented development in general. Moreover, an efficient solution for this task is particularly important given that the locations and the distribution of components and wiring harness of a specific function do not only differ between vehicle projects, but also between variants, configurations, and derivatives within the same vehicle projects. Finally, the higher the complexity of a function, the more difficult are related development, implementation and validation processes. Therefore, adequate methods are necessary to handle the high resulting complexity.

In current automotive development practice, performing the GFLT is very cumbersome. For instance, architecture diagrams as they had been introduced in Section 3.4 are limited in the type of information which can be obtained by such types of specifications. While such diagrams provide information about which parts and connections are involved in a particular function or system, they do not provide information on the spatial positions and distribution of those components and connections/wires within actual vehicle assemblies. So, function architecture diagrams do not provide a sufficient method to solve this task, or, at least, not on their own.

One possible way for obtaining information on spatial locations of function components is to contact particular experts who are responsible for specific functions or function components. However, this way can be time-consuming, it does not always guarantee acquisition of complete information and it can be prone to errors, because information have to be gathered across multiple different departments and different domains, and many vehicle functions consist of a larger number of components, sometimes 20, or even more. This problem is aggravated by the fact that there are usually frequent changes in the vehicle projects, components and implementation variants.

Another way to retrieve at least a subset of information is to use wiring harness diagrams as shown in Figure 25. Such diagrams basically do provide information on connections between components. However, a user needs to manually identify the components and connections related to a particular vehicle function, which is very time-consuming too. Moreover, while wiring diagrams may provide a rough orientation and traceability of the function architecture distribution from a top-down perspective, they do not provide any information on the exact spatial location of these components and wires in a particular vehicle project.

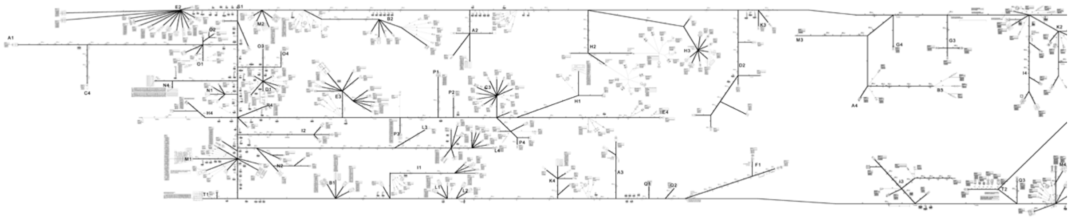

Figure 25 – Section of a hypothetical automotive wiring harness diagram (replica of an original wiring diagram); identification of function-related components and connections is a cumbersome process using this type of data due to its complexity

Exact information on spatial localization of vehicle parts can be obtained from 3D CAD data and digital mock-ups (see Section 2.1.3). However, such data does not originally include any function-oriented information so there is no connection of this parts to function architectural data. So, again, it is necessary to contact responsible experts or compare complex data tables in order to identify parts which are related to particular vehicle functions, making the task, again, very time-consuming and prone to errors.

The current lack of appropriate methods to perform GLFT-related tasks has been one of the primary reasons that motivated the research of this thesis. A primary reason is because different domains within automotive PLM such as 3D CAD based engineering and function-oriented development are still heterogeneous domains with mostly parallel development processes, systems and related data. So, these existing structures leave significant fields for further improvement of workflows and better utilization of synergy potentials. Therefore, in the further thesis contributions, a novel function-oriented 3D methodology is developed which is capable of efficiently performing the geometric function localization task.

4 State of the Art in Virtual Reality and related 3D CAD Research

This chapter explores virtual reality with particular focuses on definitions, technologies, applications and current trends. Moreover, specific characteristics and possibilities of virtual reality as a tool and medium are identified and its potential is elaborated in order to support an interdisciplinary function-oriented development.

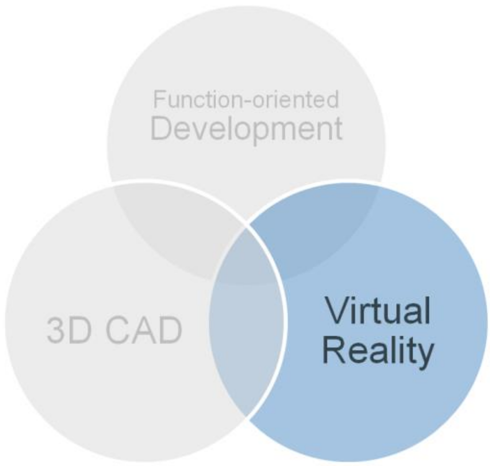

Figure 26 – Virtual reality is one of the primary subjects of this thesis

4.1 Introducing Virtual Reality

In the history of virtual reality, the first steps were already made within the early 20th century. Since then, VR development was driven by different sources like technology research, market-driven decisions and conceptual advances. A first head-based periscope display was patented in 1916 by Albert B. Pratt. Later, in 1929, a first mechanical flight simulator was developed to train a pilot at a stationary location. The first electronic digital computer, ENIAC, was developed in 1946 [76,77]. Then, in 1956, Morton Heilig developed Sensorama, a multimodal experience display system including sights, sound, smell, vibration and wind [78]. In 1965, Ivan Sutherland, who is considered as one of the pioneers in VR research, introduced his concept of the ultimate display [79]. In 1995, Nintendo released the *Virtual Boy* (originally known as VR-32) which was a 3D gaming console that was advertised to be the first portable console that could display true 3D graphics. In the year 2003, [28] described virtual reality as *"a new medium brought about by technological advances in which much experimentation is taking place to find practical applications and more effective ways to communicate."*

In the last three years, especially technical devices such as Microsoft Kinect, Oculus Rift and the HTC Vive have significantly driven the progress of virtual reality. Nowadays, computer technology and especially mobile technologies have exploded in their capabilities for computing power and 3D graphics, while prices are decreasing. 2016 can be considered as a key year which has introduced multiple advances in bringing virtual reality to the mass markets. Many developers are currently working on virtual reality applications. Today, virtual reality has entered the consumer mass market and it diffuses into different areas of application such as entertainment, engineering, crafting, medicine, architecture, education, rehabilitation and telecommunications. Summarizing, virtual reality has proceeded from being novel and poorly understood to a successfully applied technology. Moreover, advances and increases of VR applications are driving a thematic shift in research priorities from understanding VR itself towards finding and improving new beneficial applications.

4.2 Virtual Reality Definitions Review

There are multiple directions to approach virtual reality. As follows, in order to elaborate an appropriate definition of virtual reality for the context of this thesis, different definitions of virtual reality are reviewed and relevant research concepts like *immersion* and *presence* are discussed. In addition, it is argued that virtual reality can be considered as a medium, technology and tool and that it is highly capable of supporting interdisciplinary applications such as function-oriented development.

In 2012, Webster's Online Dictionary provided a definition on VR that represents a common understanding of the subject:

"A hypothetical three-dimensional visual world created by a computer; user wears special goggles and fiber optic gloves etc., and can enter and move about in this world and interact with objects as if inside it" [80].

This definition involves a three-dimensional visual world, mentions particular technical devices and includes the element of interaction.

Moving to another encyclopedic definition from 1994, the Brockhaus encyclopedia described virtual reality VR as follows:

"…the description for a computer simulated reality or synthetic world (cyberspace) that sets and interactively integrates persons with the aid of technical devices (electronic goggles, speakers, data gloves etc.)" [81].

Again, technical devices are included as well as the element of interaction. Interestingly, being different to the first definition, the term *cyberspace* is used as a synonym for a synthetic world. According to [82], any medium which encloses human communication in an electronically generated space could be a form of cyberspace.

Notably, different synonyms are used across different VR definitions for simulated and artificial environments, respectively virtual spaces, like *hypothetical world*, *synthetic world (cyberspace)* or *virtual world*. All of these terms can also be described by the term *virtual environment* (VE), which seems to be the most recognized and agreed term in virtual reality research. In this context, different types of virtual environments can be separated by its relations to real environments: projections of real environments, non-existent but fairly realistic projections and unreal projections [41]. In some literature like [83], the term virtual environment is even synonymously used for virtual reality which can lead to confusion because, in many other literature, a virtual environment is recognized as one particular element of virtual reality, next to other elements as it is explored in the following.

A recent encyclopedic definition from the Merriam-Webster Online Dictionary (2017) defines virtual reality as:

"…an artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment." [84]

This definition includes an artificial environment, sensory stimuli, the computer and a particular focus on participant's actions (interaction). References of technical devices are not included in this definition.

As it regards *sensory feedback*, many VR systems utilize visual feedback (eyes) and/or haptic feedback (typically hands). An example of a VR system stimulating an exceptionally high number of senses is the Sensorama system [85].

Furthermore, virtual reality can be defined as follows:

"…a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation" (a virtual world) [28].

In contrast to the encyclopedic definitions, the definition above describes virtual reality as a medium. Like the encyclopedic definition from 2017, this definition is not coupled to specific devices or electronic hardware [84]. Notably, this definition postulates four key elements: a *virtual world*, *interactivity*, *sensory feedback* and *immersion*.

In VR research, the term of immersion is understood as an objective measure for the feeling of a user being in a virtual environment that this environment is real. According to [86], immersion refers to what is, in principle, a quantifiable description of a technology.

Different studies indicate that higher levels of immersion and stereoscopy may contribute to a better spatial understanding of the contents in virtual environments [87-90]. Studies like those done by [91] indicate side-effects caused by immersion in virtual reality and refer to literature on motion sickness and simulator sickness. It is noteworthy that ergonomic issues are still a significant challenge even with modern virtual reality applications.

Another well-known concept in VR research is *presence.* The term presence is understood as the sense of being in an environment [92]. Moreover, presence is defined as the subjective experience of being in one place or environment, even when one is physically situated in another [93]. Moreover, the term of *telepresence* is defined as the experience of presence in an environment by means of a communication medium like virtual reality. Many literature has explored the concepts of presents like [86], [94-98].

[99] used the concepts of immersion, presence and telepresence to provide a more abstract definition of virtual reality:

"A "virtual reality" is defined as a real or simulated environment in which a perceiver experiences telepresence."

Notably, the above definition is thoroughly avoids to include technical devices, three-dimensional spaces and simulated environments.

Summarizing, primary characteristics of virtual reality include virtual/artificial environments, user interaction and sensory feedback, respectively immersion and presence. There are multiple definitions which range from considering VR as a technical system consisting of technological devices, a communication medium and, as it will be particularly explored in the later sections, a tool for industrial application.

I argue that the diversity of virtual reality's characteristics, respectively its polymorph character, is an indicator for its versatile capabilities as both tool and medium. Consequently, I propose the hypothesis that such capabilities make virtual reality an appropriate and potent solution to efficiently solve some of the interdisciplinary issues as they are required for current automotive function-oriented development. The validity of this hypothesis is proven by the contributions of this thesis. In particular, considering virtual reality and its' methods as a tool, it is explored how 3D VR methods can improve function-oriented tasks and workflows. In addition, considering VR as a medium, it is demonstrated how it can improve analysis and communication of function-oriented information.

4.3 Overview of Virtual Reality Technology

Virtual reality systems are enabled and powered by computer technology. Technological devices are necessary to enable particular elements of virtual reality experiences like *interaction*, *tracking* and *visualization*. Nowadays, a wide range of technologies is available on the market, including but not limited to cave automatic virtual environments, reality theatres, power walls, holo benches, immersive systems, as well as mixed reality technologies, haptic devices and speech systems [100]. In addition, devices such as Microsoft Kinect and Occulus Rift strife to make VR experiences more and more available for the mass consumer market [101-103]. The following sections provide a short overview on technical aspects of virtual reality.

4.3.1 Interaction

An important challenge in VR research is the development of appropriate interfaces that enable efficient and intuitive user interaction with virtual environments, virtual prototypes and digital data in general. Such type of interaction is described as a *human-machine interface* (HMI).

In many VR systems, interaction can be achieved with common computer input devices like a mouse, keyboard, gamepad or joystick. In addition, more VR-specific equipment is available, like data gloves, tracking/positioning devices, buttons, microphones and other specialized devices which suit particular applications. Interaction devices can be characterized by their accuracy, practicality and dimension while the efficiency of such devices usually depends on actual applications. For example, highly sensitive applications in the field of simulation or medical applications may require a highest possible accuracy, while a design review may have a higher priority on practicality and usability.

User interaction with digital data like such used in virtual reality applications can be streamlined and simplified with *interaction metaphors*. Interaction metaphors are sets of visuals, actions and procedures, which are supposed to increase accessibility of the interaction methodology. Usually, interaction metaphors aim to provide intuitive and easy ways of interaction which might be already familiar to users because they know them from other applications or daily life. Within a VR context, for example, such interfaces could use familiar hand gestures to assist in the navigation and handling of data within virtual environments. Moreover, virtual reality applications involve a considerable diversity of techniques for navigating through virtual environments [104]. Other works like [105] present frameworks, algorithms and techniques in order to make the application of virtual reality for virtual prototyping feasible and explore areas such as efficient interaction metaphors and frameworks, real-time collision detection and response and physically-based simulation, tracking, and virtual prototyping application development. While navigation and interaction with VR systems requires appropriate interfaces, authors like [100] point out that virtual and augmented reality itself can be considered as an interface, and, in particular, potentially the next generation type of human-machine interfaces in general.

In contrast to the real world, virtual environments do not automatically stimulate all senses. Notably, a lesser number of senses being stimulated can reduce both immersion of a VR user experience and also freedom of interaction. There are only few systems which stimulate all human senses. A rare example of a VR system stimulating all senses is the multimodal Sensorama bicycle simulator that was developed in 1956 by Morton Heilig [106]. This system stimulated all senses including sights, sound, smell and haptics (vibration and wind). In contrast, most other VR systems concentrate on visuals and, to a lesser extent, *haptics*, depending on the use case requirements.

Haptics is an interactive enhancement to virtual reality systems, allowing users to touch and feel simulated virtual objects with which they interact with. Today, it is still a very active research field to improve the immersive reality of virtual environments with haptics [107]. Haptics are especially needed in such simulations where it is crucial for the operators to touch, grasp and manipulate the virtual objects realistically as in the physical world. In industrial applications, for example, haptics can be used for assembly studies and simulation of product interaction. Haptic programming involves physical modeling, collision detection and object responses and it is still a challenging and complex task to simulate highly realistic behavior. For example, [108] discussed issues that are related to the use of haptics with virtual environments. Notably, adding haptic information such as vibration, tactile array, and force feedback enhances the sense of presence in virtual environments [109]. For example, this can improve the experience of vehicle driving simulators.

Another important aspect of interaction within virtual environments is *multi-user interaction*. For example, [110] indicate that the analysis of existing works shows that there is an important lack to be fulfilled: how to allow more than one user to interact over the same object at the same time in a fully immersive environment. Therefore, the authors developed a framework supporting the development of collaborative manipulation techniques. Notably, multi-user collaboration seems to be one of the most relevant and exclusive features of virtual reality in comparison to other types of media and there is still significant potential for further research in this area.

4.3.2 Tracking

Some virtual reality systems require information about a user's position and/or orientation within a virtual environment. For example, a multi-user virtual reality system could use avatars to show current users in the virtual world. Another example would be a system for virtual assembly training which needs to simulate the users hands within the virtual environment. Therefore, tracking technology measures a user's position and orientation and transfers it to the virtual environment. Typical devices work with mechanical, magnetic and optical technologies, but there are also other approaches. The quality of a tracking system is classified by its accuracy and occlusion, the tracked volume and noise and distortion latency. Different types of trackers are described by [111].

Tracking is a key mechanic that is usually closely coupled with interaction and visualization devices. For example, when a *head-mounted display* (HMD) is used, the position and look direction of a user's head influences what it shown by the display. Tracking is also used to improve awareness in virtual reality systems. For example, the spatial representation of users within virtual reality systems can have beneficial effects for collaborative uses. For further reading, plenty of literature is available on VR tracking systems such as [33], [112,113].

4.3.3 Visualization Technology

Typical devices which are used for visualization tasks in virtual reality systems are displays, projectors, head-mounted-displays and goggles. Notably, respective technologies are becoming increasingly available for the consumer market. Among others, well-known examples are the Oculus Rift, HTC Vive and Sony PlayStation VR. Currently, displays and projectors still seem to be used more frequently compared to HMDs and goggles, mostly because the latter devices are more prone to ergonomic issues and thus might lack in users acceptance.

A common method to increase immersion in three-dimensional virtual worlds is the use of *stereoscopic* visualization. In particular, research has revealed that stereoscopic visualization can contribute to a better perception of spatial contents and increase immersion in VR application [87], [90], [114-116]. Stereoscopic visualization technology is based on the natural human ability for spatial perception which is rendered possible by possessing two eyes. While each of our eyes captures a slightly different image of the world, both of these images are merged by our brain to provide us with a single three-dimensional image. All technologies for stereoscopic visualization use this principle by providing each eye of the viewer with a slightly different image. First mechanic devices to create stereoscopic images date back to the 19^{th} century (e.g. Kaiserpanorama). Today, many people have experiences with stereoscopic effects from 3D flip-flop images, red-green glasses and modern 3D-TVs and 3D-movies in cinemas, where either passive glasses with polarization filters or active shutter glasses are used. Notably, different technical approaches are still being worked on. For example, [121] developed a projection-based stereoscopic display for six users, which uses six DLP projectors for fast time-sequential image display in combination with polarization and programmable shutter glasses.

There are also *autostereoscopic displays* available which provide a 3D-effect without the need of additional glasses [117]. The use of specific *multi-view technologies* allows such displays to provide a spatial view for multiple viewers at the same time [118]. With currently available technology, however, there are still restrictions such as a limited number of concurrent viewers, fixed angles to perceive the 3D effect, or reduction of image resolution. For further reading on autostereoscopic displays, see [115-117], [119].

A well-known and still present problem with many VR technologies is simulator sickness [120]. Many of current VR visualization technologies, including 3D glasses, HMDs and stereoscopic displays, still suffer from ergonomic issues reported by users. For example, such issues can be dizziness, headaches or nausea, which finally lead to limited user acceptance of respective technologies.

4.4 Evaluation of an Autostereoscopic Multi-View Display for collaborative Design Reviews

Design review is one of the most prominent areas benefiting from virtual reality and immersive project technologies [122]. Also in the automotive industry, collaborative design reviews are increasingly performed with stereoscopic visualization systems to improve spatial cognition of 3D virtual vehicle models [123].

During the work of this thesis, within an OEM's electric/electronics development department, we have evaluated an autostereoscopic 40" multi-view display with 8 views in order to assess its capabilities to assist daily work flows. The intended use of the display was collaborative analysis of spatial data of 3D virtual product prototypes. The expected benefit of the autostereoscopic display in comparison to usual 2D displays was to provide a spatial 3D view on the geometric product data for all participants at the same time in order to improve the effectiveness of regular design reviews.

The autostereoscopic technology of the evaluated display is based on the approach of horizontal parallax. The display has an optical element attached to the display surface to create eight slightly shifted, perspective views. A specific display software is used to render the normal 2D image into a 3D image raster so that the light of each pixel can be properly processed by the lenticular technology of the optical element. The optimal 3D viewing distance is stated by the manufacturer to be between 3 and 5 meters. The display uses spatial multiplex technology that divides the total resolution between the multiple views. A so called c*hromatic light deflector* (CLD) sheet is used to redirect the light of each pixel to project multiple views to be viewed by the observers. To run the display, a 3D image composed of 8 different partial images is rendered by software and graphics hardware to provide 8 slightly transposed views of the scene (horizontal parallax).

We have conducted a user study that utilized the display within a realistic collaborative work scenario. The research questions of the experiment focused on the benefit of the 3D effect for perception and understandability of the 3D product data, multi-user capabilities, user acceptance and ergonomic aspects. A collaborative DMU meeting has been chosen as the experiment scenario in order to fit a representative use case. The task of the meeting has been defined as analysis and discussion of geometric product data. The virtual vehicle data used in the experiments was similar to the hypothetical data shown in Figure 27. In particular, assembly positions in a real car frontend area were discussed by the users, aiming at creating a mutual consensus across the meeting participants. The user study included 5 meetings and each meeting lasted 30 minutes. Task and goal were equal across all meetings. Each meeting involved a different group of 6 participants. In total, the experiments involved 30 attendees. The participants were experienced automotive engineers, aged between 30 and 55, and all had normal or corrected-to-normal vision.

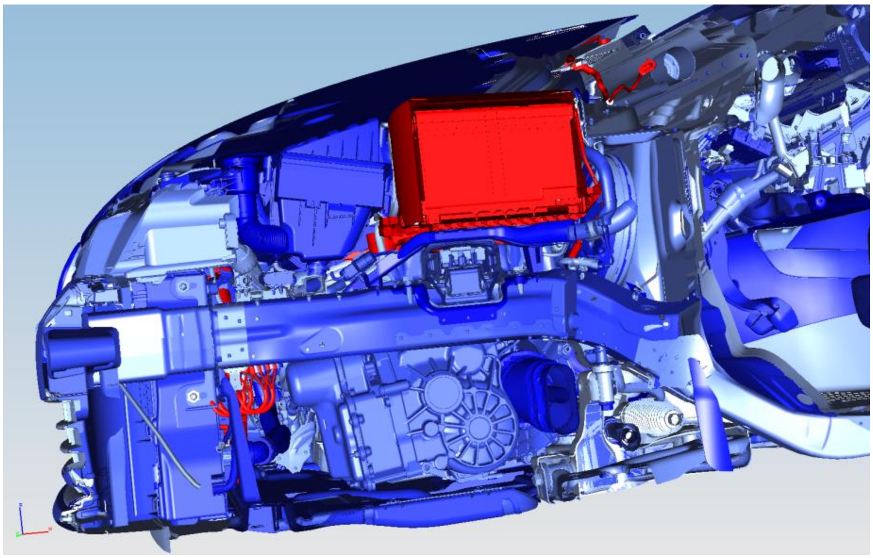

Figure 27 – Assembly analysis of exemplary car front end using a virtual prototype

During the meetings, the geometric data was displayed, zoomed and rotated by a moderator. All users were given time to analyze and discuss the content. The area of interest within the vehicle front end data was approached from different perspectives. Users were able to freely move within the meeting room, to change view angles and to acquire a decent view on the display. The discussion focused on the analysis of possible assembly issues of a specific component due to the geometric setup in the vehicle front end. A crucial aspect for this discussion was to provide all participants with an understanding of the spatial assembly situation in the vehicle front end as accurate as possible.

After the experiments, the users had to fill out a Likert scale-based questionnaire covering different aspects of the autostereoscopic display, including *image quality*, *perceived 3D effect*, *benefit of the 3D effect* for the analysis and *ergonomic perception*. The results of the study are illustrated in Figure 28.

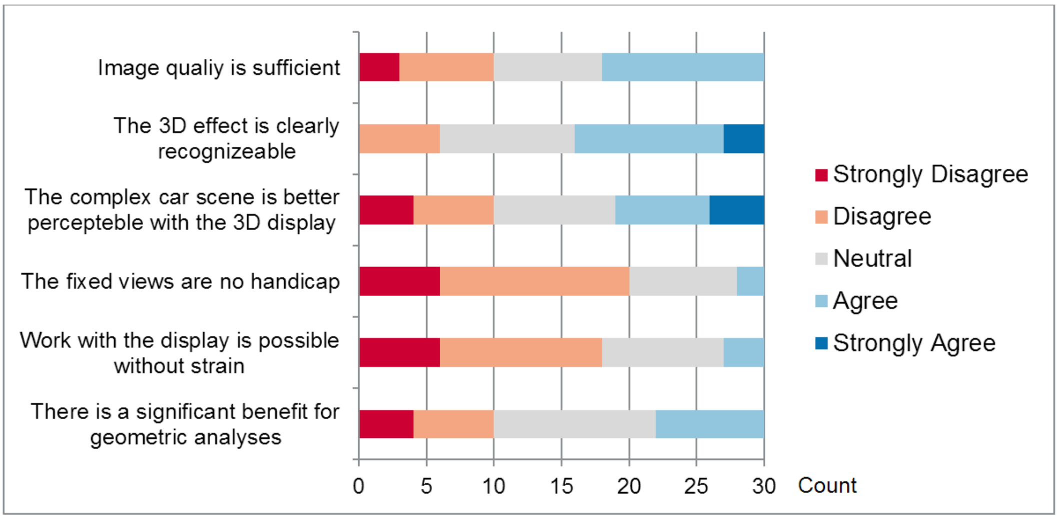

Figure 28 – User study results for use of an autostereoscopic display for collaborative design review

The study results show that perceived image quality does not meet all subjects' expectations. Only 40% of the participants consider image quality as sufficient, while 26% have a neutral opinion and the 34% remaining subjects gave a negative rating. It was notable that the image quality significantly suffers from both the lenticular sheet attached to the display surface and the reduced screen resolution. As a result, especially at lower viewing distances, elements on the screen appeared blurry and grainy. In addition, small fonts were not readable at any given distance.

There was little consensus among the participants on the recognizability of the 3D effect. 40% of the subjects have clearly recognized the 3D effect, while 26% had a neutral opinion and the remaining 34% did not report to notice a significant effect. Due to the multi-view technology, it has to be considered that the 3D effect can only be ideally recognized from specific viewing angles. In addition, even when viewing from an ideal angle, there are still a few negative effects causing parts of the image to appear slightly blurry. It has also been noticed that issues in 3D image quality become worse when the viewing distance is very low (about less than 3 meters). In addition, some participants pointed that the depth of the 3D effect was not as strong as with active stereoscopic technologies like shutter glasses.

It is well known that stereoscopic visualization improves the cognition and understanding of spatial contents [124-126]. While the results of this study provide decent support for this effect, there is still a considerable amount of users who have reported to not have recognized a significant benefit for spatial perception. After speaking with the participants, particular reasons were reported to be in the overall poor image resolution, improper viewing angles and also in differences in subjective perception of stereoscopic effects in general.

With 66%, most of the participants disagreed that the fixed and limited viewing angles are no significant handicap so there is decent evidence that this aspect hinders usability and acceptance of the state of the art of this particular technology in daily work processes. This technological matter creates the drawback that not all users are able to get the same view on the contents at the same time. This is especially problematic for the intended use case of this study because collaborative design reviews require that the viewed data is preferably equally visible for all participants in order to ensure an efficient mutual understanding and distribution of information.

As it regards ergonomic aspects, the study results show that with 60% most of the users do not feel that working with the display is comfortable and pleasant. Some participants reported about eyestrain and headache. Some users mentioned the low image resolution which feels grainy. Also, some subjects reported that the view feels uncomfortable due to the blurry appearance of the 3D effect. These outcomes provide strong evidence that this technology is considerably suffering from simulator sickness as described in [120].

Finally, only a minority of 26% of the test users sees a considerable overall benefit of the evaluated display for daily DMU applications. This conclusion is argued by the participants based on the previously discussed issues.

Summarizing, the results of this study indicate that the evaluated technology is capable of providing a decent 3D effect without the need for specific glasses and that this effect might provide a certain benefit for analyses of spatial data.

However, overall usability and acceptance of the evaluated technology suffer from significant lacks in terms of image quality and ergonomics. Especially the latter is an exclusion criterion because ergonomic factors have high priority in many industrial companies. In conclusion, the evaluated technology for autostereoscopic visualization is not yet fully convincing of being capable to assist daily work processes in the field of collaborative engineering. So, further efforts seem to be necessary to improve this particular technology. Alternatives are available due to other stereoscopic technologies like those using active and passive 3D glasses. However, it needs to be noted that such technologies are also afflicted with usability and ergonomic issues related to simulator sickness [127]. So, further research is still necessary to improve and optimize stereoscopic visualization technologies in order to better fulfill all relevant requirements such as image quality, spatial depth and ergonomics.

4.5 A Compilation of VR and 3D CAD Application Examples

4.5.1 Non-automotive applications

Many VR applications are settled in the crafting and manufacturing industries like automotive, aircraft and shipbuilding. However, there are also multiple other domains that utilize its potentials. In fact, the first professional applications of virtual reality are to be found in the military domain that started to use VR for simulation, education and training purposes. Typical military applications are combat simulations like flight simulators which are used for the training and education of soldiers and pilots. For further reading, some recent examples of military VR applications are provided by [128-130].

In the International Space Station NODE 2 Program, [131] evaluated DMU as a tool to support verification, integration and testing of spacecraft. The authors found that DMU helps to decrease development and manufacturing costs and enables detection of problems at early development stages and eliminates the need for real physical mock-ups. Based on the positive results of their evaluation, the authors indicated that DMU will be used for the design, support, assembly and production of new spacecraft in future.

In the medical sector, VR is today used in multiple areas like minimally invasive medicine, surgery, stereo endoscopy, angiography, plastic surgery, microscopy and other applications. For example, virtual and augmented reality offer great potentials for medical education, diagnostics, operations and intraoperative assistance [132]. Moreover, medical applications of virtual reality include surgical simulators, telepresence surgery, complex medical database visualization, and rehabilitation [133]. [134] show how current technologies in virtual reality and motion-tracking can be used to support physical rehabilitation with a game-based approach. As it regards neuropsychology, an analysis of various assets for VR applications is provided by [135]. Works like [136] and [137] show potentials of VR to better deal with particular challenges of posttraumatic stress disorders.

A considerable influence on the advance of virtual reality is exerted by the entertainment industry. Video game consoles are offering virtual worlds, multi-user interaction, movement detection and haptic feedback since multiple years. Furthermore, the gaming industry is an important factor in driving the capabilities of graphics hardware and VR devices. Well-known examples are the Microsoft Kinect tracking sensor and the Oculus Rift head-mounted display. Since 2010, stereoscopic displays are broadly available to the mass consumer market. Such displays contribute to increase immersion in virtual reality experiences due to spatial visualization [138]. Moreover, due to technological advances, the distribution of augmented reality applications is rapidly growing in hand-held devices like smartphones and tablets [139].

A research work from an archaeological application has been presented by [140]. The authors developed a prototype for collaborative application that allows archeologists from remote locations to interact in real time with 3D archeological models through a shared virtual environment using tele-immersive technology. A stereo reconstruction algorithm enables accurate real-time capture of the uses within the 3D environment. The authors pointed that awareness in collaborative systems may arise through the visibility of the effects of actions on the objects of work, and the imitation factor concerns the capacity to create mental maps of someone's actions. The authors argued that such effects enable beneficial new ways of research, training and communication in archaeology. Notably, research on collaborative use of 3D data is also an interesting field of future work as it regards the subject of this thesis.

Virtual reality also is a common tool in architecture and tourism management. In the architectural domain, work like [141] show that virtual technologies enable decisions on design and virtual walkthroughs through buildings that are still in the planning and conception phases. In the tourism sector, an approach by [142] shows how virtual reality can effectively solve problems in tourism management teaching.

[143] provide an interesting research that shows how virtual reality can be used for 3D simulation of ship motions to support the planning of rescue operations on damaged ships. [144] propose a work on ship evacuation analyses based on human behavior algorithms to simulate real world evacuation scenarios. For further read, a collection of efficient VR uses in the ship building industry is presented by [145].

Within the aircraft maintenance industry, [146] evaluated virtual reality as a tool to support the training of aircraft inspectors, with particular focus on its capabilities to improve visual inspection tasks. The authors found that VR offers a high degree of presence and is superior for the training process in contrast to the traditional PC-based training tool. [147] proposed a virtual collaboration environment for aircraft design and [148] show a system that is capable of efficient interactive rendering and modification of huge aircraft 3D models within a VR environment. [149] provide an interesting work that proposes a generic virtual reality framework for maintainability and assemblability test of complex systems. The authors successfully demonstrate the capabilities of their framework on an aircraft carriage and a railway coach cooling system.

A comprehensive survey and evaluation regarding virtual reality research in the manufacturing industries dating from 1992 to 2014 is provided by [8]. Notably, as it regards an integration of CAD and VR, the analyzed literature mostly deals with aspects such as design review or simulation. However, there is no research which focuses an integration between CAD, VR, function-oriented development and facilitated methods.

4.5.2 Automotive VR Applications

The automotive industry is one of the key drivers in professional applications of virtual reality and virtual prototyping. According to [24], the complexity in the automotive industry is not to be handled without simulation any more. [150] indicated that the importance and relevance of virtual technologies for industrial and general application has continuously increased over the last ten years. Moreover, [151] note that methods and tools are needed to verify development results which can be used early in the product development process. As a result, virtual prototyping and applied virtual reality methods have become major tools to enable analyses, simulation and reliable decisions at early stages of development. Such technologies help to increase product quality as well as to reduce development costs and times [152]. In the automotive industry, typical applications of virtual technologies include design reviews, assembly studies, behavior simulation and ergonomic analysis and there is much literature substantiating such increasing applications. For instance, [153] discuss different examples on the application of virtual prototyping in the automotive industry. The authors emphasize that close collaboration between industry and research is necessary to handle the challenges of virtual prototyping. In addition, an overview of VR-techniques for industrial applications is provided by [41]. Studies like [110], [154] and [155] indicate that virtual technologies are highly qualified for supporting collaboration and communication, which are crucial requirements for a successful product lifecycle management. Moreover, for example, [156] utilize mixed reality technologies to allow a closer collaboration among designers, final users and engineers and hence reduce the time for reviewing and validating car interior designs. So, the results of the above research works provide further indicators for the suitability of VR to support a function-oriented development which its high requirements at interdisciplinary collaboration and communication.

[100] identify applications of VR like design analysis as well as visualizing post-processed results of simulations, such as DMU analyses and scientific visualizations of fluid dynamics and crash analysis results. [157] investigate the application of VR for virtual prototyping to verify assembly and maintenance processes and point out the increasing relevance of VR as a tool for virtual prototyping. The authors emphasize the importance of a seamless and complete integration of VR into the existing CAx and IT infrastructure to become a wide-spread tool. [158] utilize virtual and augmented reality applications to support manufacturing processes.

[159] introduce *Smart Hybrid Prototyping* (SHP) as a solution for multimodal and interdisciplinary evaluation of virtual prototypes by integrating concepts and technologies of different fields like virtual, mixed and augmented reality, DMU, HMI and simulation.

[150] provide a comprehensive volume and overview of virtual technologies and industrial applications, based on the AVILUS project. The AVILUS project was a 3-year joint research project focusing on 42 research scenarios, involving large and medium scale industry, universities and research facilities. The goal of the German government funded AVILUS project was the application and user oriented research, development and evaluation of future oriented technologies in the context of virtual and augmented reality [160,161]. Research results show beneficial applications of virtual reality in different fields such as PLM information management, simulation and rendering, tracking, interaction and recognition of geometric data. A similar research project focusing on augmented reality has been introduced and exposed in [162].

[29] provide a research work which involved reviews of relevant VR literature as well as 11 interviews of engineers of a world-leading automotive OEM. The authors argue that the automotive industry benefits from increased usage of VR for optimizing products and development processes. Their work shows that VR can improve several aspects of the product development process and that continuous reviews of existing systems and studies regarding new opportunities are required to exploit potentials for increasing competitiveness and product quality.

Such works further encourage the approach of this thesis which focuses on the revelation of new opportunities and exploitation of synergies due to an increased utilization of virtual reality methods. In particular, the work of this thesis extends those fields of previous and current VR research and application by adapting it to improve relevant tasks and workflows for function-oriented automotive development.

4.6 Elaborating Capabilities of Virtual Reality as New Media

Three major trends exist that drive the technological evolution in the modern knowledge society: an increase of richness and completeness of communications towards immersive virtual telepresence, an increasing relevance in the role of mobility aspects, and pervasive diffusion of “ambient intelligence” through omnipresent network technologies like cloud computing [163]. In that regard, I argue that virtual reality is an enabling technology for which still more research is necessary to go beyond understanding the technology itself and to advance the understanding of how to properly use it. Especially the latter challenge raises multiple research questions, including but not limited to proper ways of communication, ergonomics and social behavior. The heterogeneousness of these questions emphasizes the necessity of interdisciplinary approaches of future research for virtual reality and other novel media technologies.

[28] argue that virtual reality has many features that combine to make it a *"truly unique medium"*. These features include the ability to manipulate the sense of time and space, the option of interactivity, multiple simultaneous participants and the potential for participants to drive the narrative flow of the experience. Other significant potentials of virtual reality include mixed sensory stimulation and the element of immersion. This diversity requires an understanding of virtual reality as a complex, multidimensional medium. Consequently, particular VR applications raise research questions that are not entirely addressable by one scientific field on its' own. For example, when considering collaborative VR systems with multiple users, particular research questions require such areas as media sciences, communication sciences and social sciences. Moreover, certain issues might require an interdisciplinary association of these fields. For instance, one scientific field of research that had spawned from the necessity for interdisciplinary research is computer supported cooperative work (see Section 3.1.9). Notably, computer supported cooperative work is not limited to virtual reality but generically exposes the use of technology in support of human collaboration.

The necessity for interdisciplinary approaches in VR research is further confirmed by such studies that explore virtual reality as a communication medium rather than regarding it as a technical system. For example, [164] investigated virtual reality as a tool and medium for communication, [165] considered possible implications of virtual reality for communication research and [99] provided a definition that enables classification of VR in relation to other media. Moreover, there is a need for studying VR not only as a human-machine interface, but in terms of content and style like traditional media such as novels, cinema and television [166].

A common theory in communication sciences is the *channel reduction theory* [167]. Considering virtual reality as a highly multidimensional medium, this theory can be adapted to virtual reality. The channel reduction theory states that computer-aided respectively technical-mediated communication is inferior to face-to-face communication because important inter-human communicative aspects like facial expression, gesticulation and/or environmental aspects are lost. In particular, the theory describes a sacrifice of communicative aspects in different layers [167]. It criticizes that in a technical-mediated communication many senses are not stimulated during the interaction with the dialog partner (e.g. view, smell, feel). Moreover, emotions are not automatically transported and therefore may not be noticed. In addition, it describes a loss of the natural conversational context and the personal characteristics of the dialog partner. Finally, especially in asynchronous communication, there can be differences in time and space. Summarizing, the channel reduction theory speaks of an "alienation of communication" which may unsettle the reference to reality.

From another point of view, the channel reduction theory is not fully convincing in terms of communication sciences and, at least, can be significantly put into perspective because the deficits of the theory are compensated due to a changed communication behavior [168]. For example, today, many modern societies, organizations and even daily family live are barely thinkable anymore with face-to-face communication only. Instead, modern media- and computer-aided communication extends the range and accessibility of communication and adds new communication channels as well (i.e. mobile phones, internet, etc.). Though, one the one hand, technical communication can reduce the richness and the way in which communication is happening, one the other hand, many new channels of communication are initially enabled by technical communication. Notably, in particular situations like business communication, it can be argued that the use of reduced or compressed information can even be of advantage for a communication because it can help to avoid redundancies and misunderstandings.

Applying these discussions to virtual reality, I argue that VR applications have considerable potentials to overcome many of the limitations of the channel reduction theory. For instance, a user whose movement is tracked and transmitted by a sensory device is able to transfer the elements of spatial positioning, gesture and even mimics into a computer-aided communication. In that regard, this potential makes virtual reality superior to traditional types of media like telephone, television or print media because VR systems are able to provide enriched communication by maintaining particular communication layers which are lost in other types of media. Naturally, these capabilities are increasing with further technological progress. In support of this argument, for example, the research work of [169] explores *3D teleimmersion* and points that it is an emerging technology that enables users to collaborate remotely by generating realistic 3D avatars in real time and rendering them inside a shared virtual space. The authors noted that the use of teleimmersive environments can provide benefits for collaborative work on 3D data such as medical imaging, scientific data, archaeological data, architectural or mechanic designs or remote training and teaching. As it regards technology, the authors presented an approach using image-based stereo and Kinect and discussed current challenges regarding capture, transmission, rendering and interaction.

Another research work on tele-presence has been proposed by [170] who presented a wall of multiple LCDs and cameras which the authors call the *Extended Window Metaphor*. The authors identified technical constraints in order to provide a sufficient user experience and demonstrated the feasibility of VR communication which already gets very close to real face-to-face communication. As it regards appropriate technology, [171] presented different strategies focusing on appropriate camera selection, data preparation and compression in order to efficiently transmit huge amounts of data as they are required for high-resolution telepresence systems. In addition, [172] provided a research work on collaborative virtual assembly tasks where haptic feedback is used to transfer information about a partner’s gestures during closely coupled collaborations. Their results provide evidence for a correlation between applied forces and a level of efficiency as well as general improvement of gesture coordination in collaborative VR applications due to haptic feedback.

Summarizing, there is still much space to further improve technological capabilities in aspects such as ergonomics, human-machine interfaces, realtime data synchronization and visual quality. Nevertheless, however, current media technology is already capable of providing virtual reality experiences fulfilling all primary requirements claimed by scientific literature, including interactivity, sensory feedback, virtual worlds, immersion and multi-user capabilities. Moreover, virtual reality is able to overcome weaknesses of traditional media in terms of richness of communication. For example, VR is capable to provide both remote visual communication of product data and face-to-face communication across collaborators. Thus, in support of the hypothesis depicted in Section 4.2., I argue that the richness of communication that can be achieved with virtual reality systems can be a significant benefit for complex, interdisciplinary tasks like such which are required for a function-oriented development.

4.7 Research Trends, Gaps and Potentials

In this section, different related research works are reviewed in order to identify current research trends, gaps and potential areas for future work, and also to provide a context for the particular research scope of this thesis.

In the previous sections, it has been explored that applications of 3D CAD, virtual reality and virtual prototyping can be beneficial in multiple ways such as improving product quality, reducing development costs and times and enabling decisions at earlier design states. However, respective approaches concurrently generate challenges by themselves. For example, [173] indicated that automotive design processes involve many collaborators and several CAD systems within a distributed system topography. In fact, today there is a significant diversity of heterogeneous systems, domains and collaborators that are involved in related development processes and multiple separate tools are utilized to address different requests of the PLM like design, simulation or visualization. For example, one system might be utilized for the initial construction of geometric CAD-data and another system is used to properly store the data for configuration management. Moreover, a DMU analysis based on this data might be conducted in a third system while a fourth system is eventually used for high-end visualization. This example already includes four different systems which only focus on geometric product data and do not yet include any of many other relevant product properties.

As of today, many of such different domains, systems, applications and data sets are not yet fully connected and integrated. For instance, [9] pointed to the issue of *domain isolation* and the resulting need to overcome related trans-area barriers. For instance, in different domains of automotive development, different product data like geometric representations or domain-specific product features are utilized. This data may or may not be significantly interdependent and reciprocally available in other domains. However, especially such scenarios like a function-oriented development involve a considerable amount of interdisciplinary, collaborative work with a high share of interdependencies. In such scenarios, a reciprocal connection of data across different domains obviously provides a significant potential to improve efficiency of collaborations and tasks which require cross-domain information. This finding actually has been a primary starting point for the research work of this thesis and the thesis aims at providing clear evidence for such synergy potentials.

Moreover, many practical applications of virtual prototyping still focus on particular aspects of the virtual product model, like geometry, particular simulation models, etc. In contrast, a complete virtual prototype would include all data and information that is relevant to the respective product. Already in 1999, [157] noticed that a lot of the data needed for a complete digital mock-up were just not available yet, such as process information, material properties, etc. In contrast, a look at the current state of CAD and virtual prototyping shows that despite many progresses even current virtual prototypes are still far away from being complete virtual prototypes. So, in 2017, [7] stated that *„future-oriented CAD models will additionally include complete data structures, providing all product and process related information for effective cross-domain development of mechatronics systems“* and *“[...] this requires a further integration of product characteristics into 3D models […].”*

Examples of approaches to improve development processes based on integration concepts can be found in the fields of both digital and functional mock-up. For instance, solutions like such proposed by [173] show how challenges of heterogeneous, collaborative CAD assembly can be handled by DMU approaches. Approaches like [174] and [40] assist in streamlining interfaces between CAD data and virtual reality applications to benefit in design review processes. Moreover, the FMU framework proposed by [175] helps to shorten development times of multi-domain systems and allows integration tests at early stages of development. [176] use a wireless, real-time transmission to transfer simulation data to a rich 3D environment creating a comprehensible visualization of such data. Thus, their work assists in validation and presentation of simulation data, especially for non-exports. In addition, research works like [177] emphasize the need for interdisciplinary cross-skill engineering collaborations. In addition, [178] proposed an interdisciplinary approach to functional prototyping and the authors present a framework that couples different simulators to address the necessity of timely simulation, review and debugging of multiple mechatronic components in complex automotive systems. The results indicate that the author’s work allows early functional design reviews and assists in the specification and evaluation of mechatronic systems. These works are examples that further approach the use of integration concepts towards a more complete and sophisticated virtual prototyping.

In support of mastering the challenges of complex automotive systems, many research works also focus on improving accessibility of information due to visualization. For instance, [179] provide a dual-view visualization for exploring functional dependency chains of in-car communication processes. One view focuses on hardware component dependencies using a space filling approach while the second view uses an interactive sequence chart to displays functional correlations. In addition, [180] have proposed a visual tool for exploring and communicating an automotive bus technology to support automotive engineers in the development of car communication networks. As a result, they found beneficial application in utilizing new methods for information visualization in a complex domain, in which the only access to data was textual so far. In addition, [181] developed a system for visualizing spatial sensor data to assist in the development of automotive driver-assistance systems based on environmental perception. Notably, this thesis fits into such as the above works in the sense that it also enables new methods for 3D visualization of function-oriented information.

Summarizing, recent related research works demonstrate the potentials of novel approaches to data integration, analyses and information visualization to assist in virtual prototyping. In many cases, such works are based on cross-system solutions and interdisciplinary interfaces. Moreover, recent proceedings indicate a trend of further advancement towards more holistic and complete virtual prototyping, incorporating an increasing amount of virtual reality methods, product information and synergies. However, it can also be noted that there are still issues and significant potentials to further improve processes, systems and workflows. Thus, both the promising results of such related works and currently remaining challenges and research gaps further encourage the approach of this thesis which applies 3D CAD and virtual reality methods on function-oriented development. So, the interdisciplinary data integration concept which is proposed in this thesis particularly fits into the depicted trends by connecting heterogeneous development domains and related artifacts in order to explore synergy potentials and to enable beneficial new methods for product lifecycle management.

5 A Concept and Prototype for Integrating Automotive Function Architectures with 3D CAD Models

This chapter proposes an overall concept for an interdisciplinary and consistent integration of automotive function architecture data with 3D CAD data, including a prototypical implementation. Hence, this research focuses an interdisciplinary, first-time connection between two automotive domains, 3D CAD based engineering and function-oriented development, which are currently mostly run in parallel processes as illustrated in Figure 29. In particular, this contribution includes identification and definition of basic requirements for a consistent data mapping of automotive function architecture data and 3D CAD data. Moreover, I propose a self-developed XML-based data format that allows for system-independent description and exchange of function-oriented data and which enables a prototypical data integration within established systems. Consequently, based upon the previous steps, I develop a prototypical implementation of the data integration concept. With this prototype it is made possible to develop and research novel function-oriented 3D methods to improve multiple relevant tasks within a function-oriented development as described in the subsequent Chapter 6.

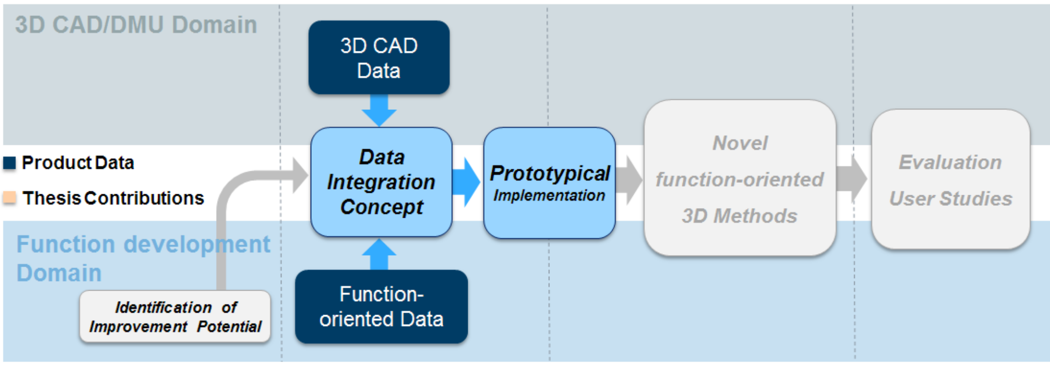

Figure 29 – 3D CAD-based engineering and function-oriented development are currently still running in heterogeneous and parallel domains. The thesis integration concept aims at enabling an interdisciplinary connection between these two domains

5.1 Requirements for Consistent Function-Geometry Mapping

The data integration that is approached in this thesis aims to enable a connection between two types of data: function architecture data (see Chapter 3.4) and geometric CAD data (see Chapter 2.2.1). This connection requires an assignment (*mapping*) of elements of the function data to corresponding geometric parts of the CAD data (see Figure 30). In particular, *consistent* mapping is understood as a *complete* and *explicit* data allocation or assignment, which is discussed in the following.

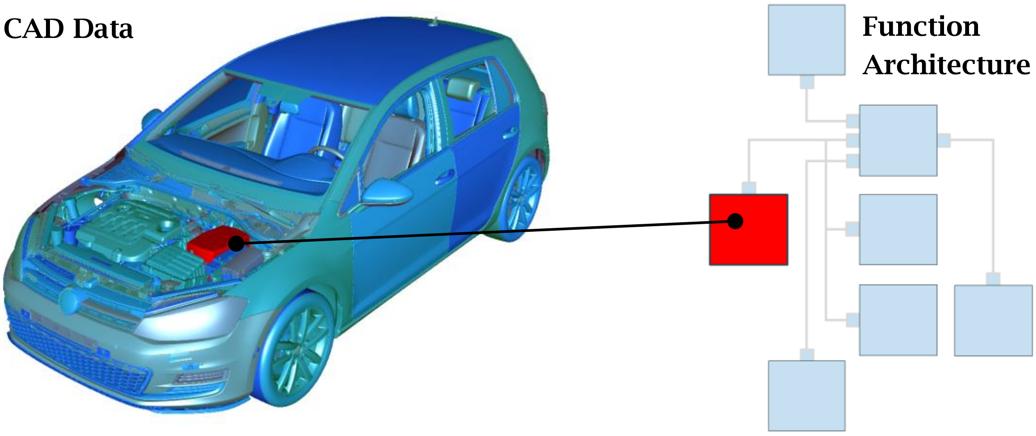

Figure 30 – Consistent data mapping addresses the proper association of function architecture elements with geometric parts of the CAD data

5.1.1 Allocation

A primary question of the intended data mapping is how to make particular elements of function-oriented data *know* to which geometric parts of the geometric data they are related to. In terms of data structures, the technical answer to this question is the use of clearly defined *identifiers* which, for example, can be implemented as explicit key fields to control data allocation. Consequently, such identifiers need to be available in respective data formats and systems, and maintenance of such identifiers needs to be considered in design and implementation of related processes. In order to ensure logical correctness of the mapping, this step requires accurate information about the mapping on function elements to particular geometric parts. This task can be a particular challenge because it might require a high degree of expertise, cross-domain exchange of information, adequate tools and processes, and periodic data updates due to the quantity of iteration cycles during automotive development. For practical process implementation, maintenance of such identifiers should be automated as much as possible to improve the workflow and to minimize errors.

5.1.2 Completeness

An obvious yet important requirement for a consistent data mapping is a sufficient integrity and level of *completeness* of the data. In particular, in most cases, it is desirable and necessary that all elements of the function data can be associated with geometric parts of the CAD data. Regarding this basic requirement, it needs to be considered that CAD data and function data do not necessarily involve the same level of information. For instance, if a set of function architecture data contains variants, a complete mapping of this function data requires a set of CAD data that includes these variants as well. If this is not the case, a complete mapping of all variants is not possible, and, for example, it might only by possible to map a single variant with the CAD data. However, in some cases, this could still be acceptable though as long as the information is sufficient for the particular use case. Nevertheless, for any data which is relevant for a use case, it needs to be prevented that function data cannot be properly mapped because of missing geometry or missing granularity of geometry, which leads to the next requirement of *sufficient granularity*.

5.1.3 Granularity

Sufficient *granularity* is identified as another basic requirement for a consistent data mapping and can be a prerequisite for the completeness of the mapping. *Sufficient* granularity means that a proper mapping of function data requires a level of granularity that is sufficient to assign all function elements to related geometric parts. This requirement is particularly crucial for the mapping of function connections because in the function architecture data a connection is usually represented as a single element (e.g. a line within an architecture diagram). In the 3D CAD data, however, a connection is represented by wiring harness, usually consisting of multiple wire *segments*. Thus, the data mapping results in a one-to-many or even many-to-many relationship because it can be necessary that multiple elements of the function data must be mapped to multiple geometric parts (wire segments). In practice, granularity of the geometric CAD data can especially become an issue if geometric parts are not atomic but grouped and merged as discussed in Chapter 2.2.1. In this case, an explicit separation and allocation to the individual wire segments is not possible. Thus, a sufficient granularity of CAD data needs to be ensured to enable a complete mapping.

5.1.4 Relations

Logical requirements for the mapping are derived by the definition of possible *relations* between function elements and geometric parts of the CAD data. Basically, the relationship between function elements and geometric parts is *many-to-many*. For example, it can be possible that multiple elements of the function data need to refer to a single geometric part. For instance, an instrument cluster physically is a single vehicle assembly part so chances are that it is represented in the CAD data as a single geometric part as shown in Figure 31, left side. In the function data, however, from a function-oriented point of view, it is possible that the instrument cluster is represented by two separated components, a controller and an actuator as shown in Figure 31 right side. In this case, both the controller and the actuator map to the same geometric part of the CAD data.

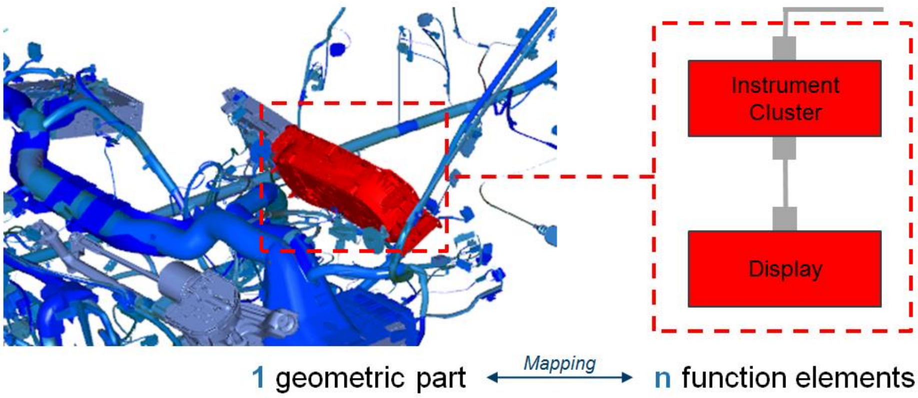

Figure 31 – Mapping of one geometric part in the CAD data to 2 components in the function data

On the other hand, it can as well be necessary to map a single function element to multiple geometric parts. For example, a single connection within a function architecture might need to be mapped to multiple parts (wire segments) of the geometric data as shown in Figure 32.

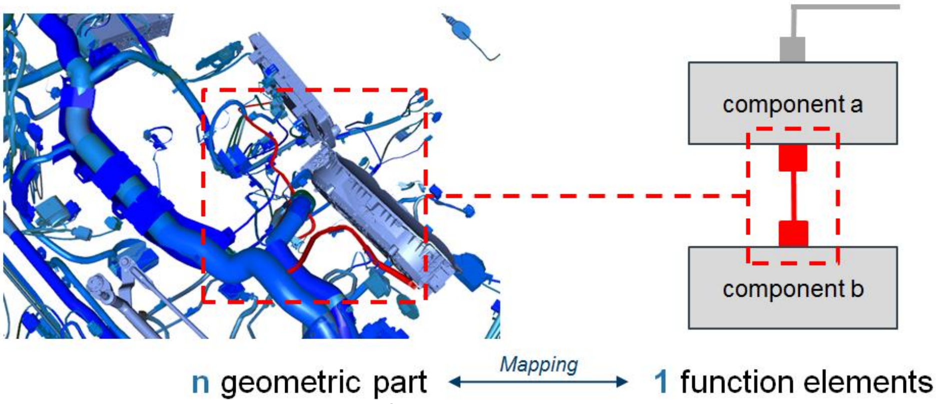

Figure 32 – The connection is composed of many geometric parts in the CAD data but maps to one connection element in the function data

As it concerns elements of the function architecture data, relationships can be differentiated between components and connections. A function component will usually be mapped to one geometric part of the CAD data. A connection would be mapped to one or more geometric parts, unless the connection is within a component and thus has not geometric representation in form of a geometric wire segment.

5.1.5 Summary

Summarizing, basic requirements for a consistent data mapping can be listed as follows:

- **Allocation.** Unique identifiers are necessary to enable an explicit assignability of all data elements that are to be mapped.
- **Completeness**. The data volume of the CAD data needs to include all geometric parts which are required to completely assign all configuration-relevant function elements.
- **Granularity**. A sufficient level of CAD data granularity is required for a proper data assignment, especially for the mapping of connections/wires. Insufficient granularity can lead to insufficient completeness.
- **Relations**. Proper relationships need to be supported and ensured. Basically, relations between elements of the function data and parts of the CAD Data can be many-to-many. Consistent mapping requires:
    - Each function component needs to be mapped to one geometric part for a particular car configuration.
    - Each function connection needs to be mapped to at least one geometric part unless the connection is a controller-internal connection.

The above requirements are basic and minimum requirements. Depending on use cases and scenarios, additional requirements might be needed.

5.2 System-independent Description of Automotive Function Architectures

Multiple systems are available on the market for system architecture design and also for CAD and 3D data utilization. As it has been exposed in Chapter 4.7, the heterogeneity of tools can be afflicted with particular issues like insufficient capabilities for data exchange between tools (e.g. loss of information, cumbersome workflows) or the need for additional interfaces both technically and process-wise. Some technical barriers for an integration of function data with geometric CAD data can be reduced by making function-oriented data available in a system-independent format. So, in this subsection, I propose the XML-based format which I have developed to enable a system-independent and interdisciplinary description of automotive function architectures. This format is used for the prototypical implementation and it provides a solution to easily exchange and utilize function data across different systems and domains of automotive development.

Technically, other appropriate data formats can be used as well. However, I have chosen to use the XML format (see Chapter 2.2.3) because XML is a widely distributed and common format that can be easily used for system-independent description of data. It exploits advantages such as language- and platform independence, human-readability and easy validation using schemas. Additional arguments in support of using this format include that many engineers in the engineering working domain supporting this thesis were already familiar with XML and it also provides the option, if necessary, to add/change data per hand without the need for additional tools. So, for the requirements of this thesis work, XML provides an efficient and straight-forward solution for system-independent exchange of function-oriented data.

Based upon the mapping requirements (see Section 5.1) and the properties of the function data, basic prerequisites for the proposed XML schema can be identified as follows:

- Identifiers shall be used to enable an explicit data allocation.
- A function shall be a set of its elements, including components and connections.
- Components shall be of different types (including at least actuator, sensor, controller).
- Connections shall be of different types (signal-wire, can-bus, and others).
- Functions shall comprise 1…* components and 0…* connections.
- Function elements and CAD parts can have many-to-many relationships
- Mapping references shall be optionally specifiable in XML files.
- Support for configurations/variants is required.
- Additional properties shall be optionally specifiable for function elements.

In the following, individual steps in the development of a custom XML schema based on the above requirements are described. The schema implementation is documented using visual diagrams.

The schema is supposed to support the descriptions of single functions as well as multiple functions in a single XML file to provide a higher grade of flexibility. Therefore, a data element called *Function* is defined to include the function data and in addition, a parental object called *FunctionCatalog* is defined which can include any number of Functions as shown in Figure 33. Both elements the *Function* and *FunctionCatalog* are declared as global elements so that they both can be used as root elements in XML documents which are both considered as valid. The Function element is specified by a complex type *FunctionType* that will be specified later.

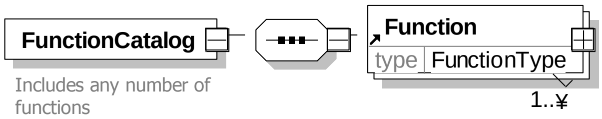

Figure 33 – The schema shall support multiple functions per file so a function catalog is defined

The data mapping requires functions and all of its elements to be referable by explicit identifiers. In addition, elements shall be given a descriptive name to increase human-readability of the data. Finally, functions and its elements should be optionally relatable to particular vehicle configurations. This schema implements the above requirements by providing respective data fields in a global attribute group called *Header*. This data structure can be reutilized by all data elements of the schema (see Figure 34).

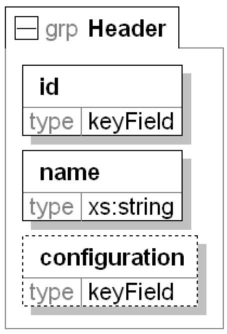

Figure 34 – Header Attribute Group

The schema uses a complex type called *ComponentType* that defines the data type for the description of function components. This data type uses a simple type *MechatronicType* to define the type of the component. Enumerations are used to restrict values to either *actuator*, *sensor* or *controller*. An optional element *Annotations* is added for additional descriptive information. In addition, an element called *MappingLink* is included which is used to refer to a geometric part to which the component is mapped to. This element is defined as optional because XML documents without mapping links should also be allowed as schema-valid documents. Finally, another requirement for the schema is the optional extensibility of function elements with additional metadata. For example, such metadata could include additional information like the software version or test maturity of controllers. Since the type of required metadata is significantly depending on particular functions and applications, the schema uses a generic data structure to provide a sufficient grade of flexibility. Therefore, a complex type *MetaDataType* is defined which uses the attributes *DataField* and *Value* to describe the name and the value of the metadata element. A function element can utilize any number of *MetaData* elements to enable the inclusion of additional information. Including the above data elements, the final data type for ComponentType is implemented as follows, shown in Figure 35:

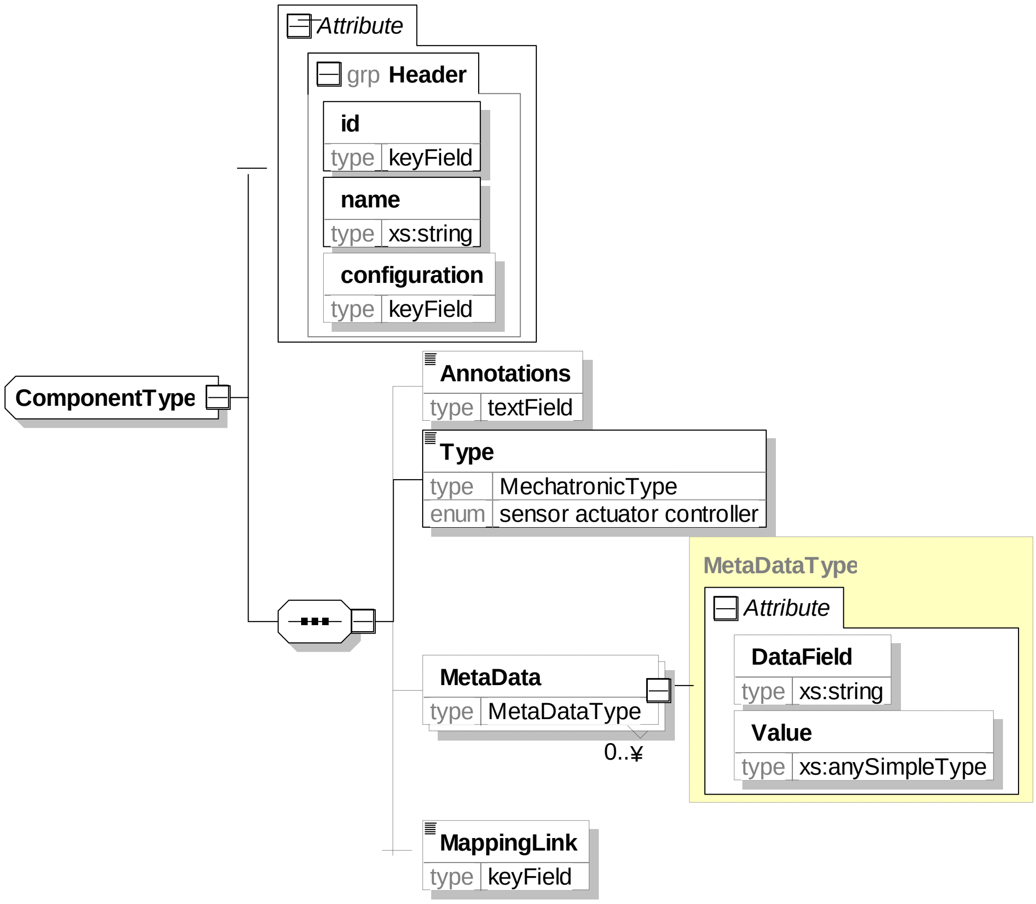

Figure 35 – Component XML model

The implementation of function connections is similar to those of function components and uses a complex type *ConnectionType*. This data type also includes the *Header* attribute group to provide an identifier, name and configuration key. Annotations, however, are omitted in favor of simplicity because they are not needed for actual use cases. *ConnectionType* includes a simple type *WireTyp* to describe the type of connection and uses enumerations to restrict its values to either *can-bus*, *signal*, *ground*, *supply* or other. In addition, an element called *Connector* is defined. This element is needed to create the logical connection to the components to which the connection is attached to using a single identifier. Minimal occurrences of connectors are set to 2 because a connection needs to be consisting of at least two connectors. Analogously to components, the optional elements *MetaData* and *MappingLink* are included. However, the *MappingLink* is set with unbounded maximum occurrences to allow a connection object to be assigned to multiple wire segments. Figure 36 illustrates the complete *ConnectionType* data structure.

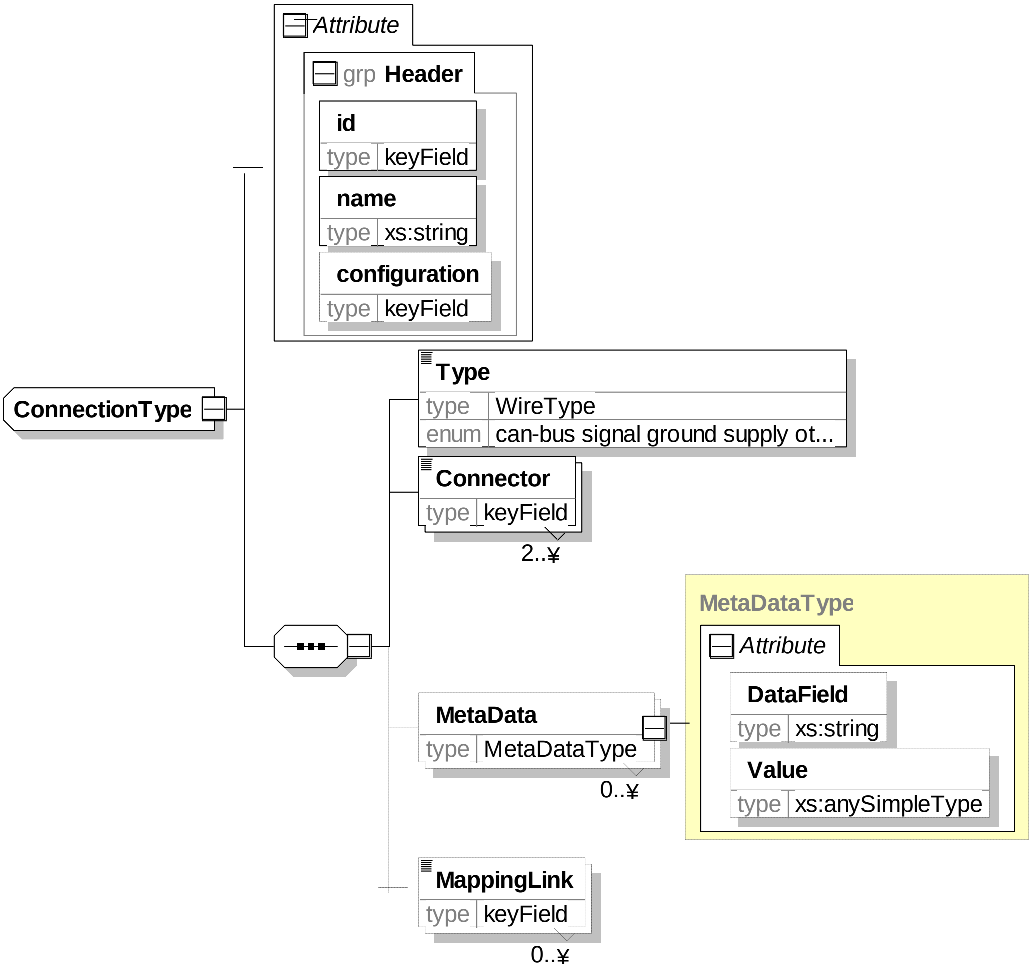

Figure 36 – ConnectionType element

The complex data types *ComponentType* and *ConnectionType* are implemented in the XML schema as sub elements *Component* and *Connection* of the data type *FunctionType* that defines the primary data structure for functions. The Connection element is considered optional (0…*) because a function theoretically can be consisting of one controller only without any further connections. In addition, the *FunctionType* element also uses the *Header* and *Annotations* for same reason as the other elements, as shown in Figure 37.

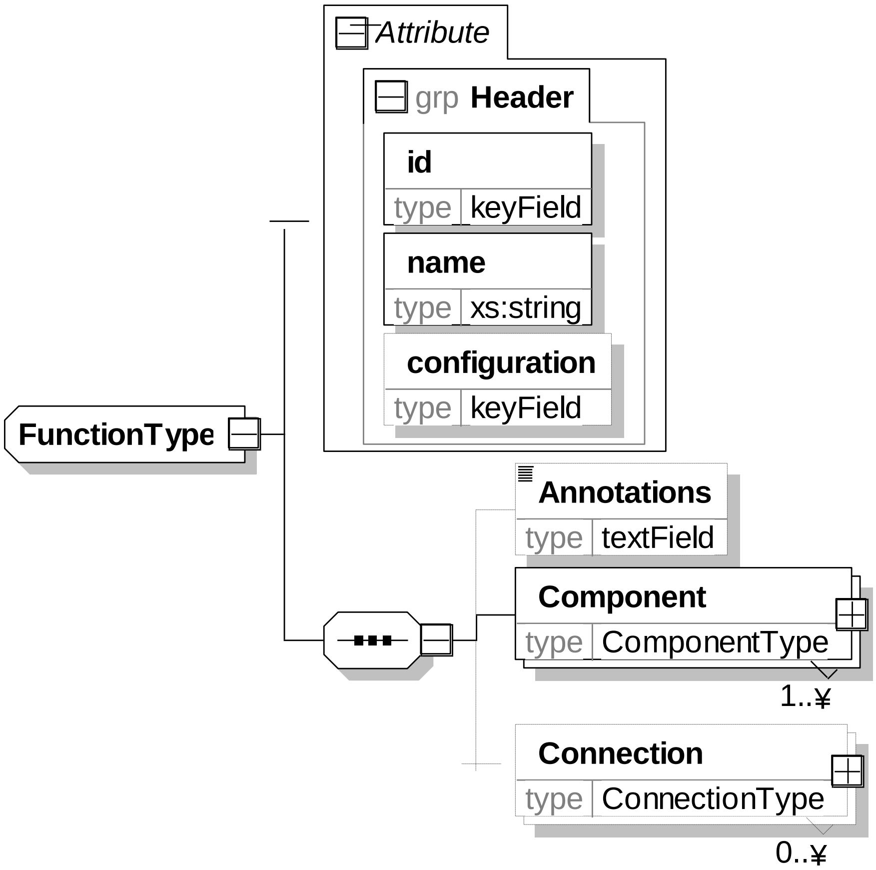

Figure 37 – FunctionType XML description

An alternative implementation that also has been considered in the development of the schema is the use of an additional parent element *FunctionElement* that provides a generalization for the elements *Component* and *Connection* (see Figure 38) and uses the concept of inheritance as it is known from object oriented software engineering. In this schema, however, a global attribute group is preferred for the reutilization of shared attributes because it avoids an additional level of hierarchy in order to achieve a lower overall complexity of the schema.

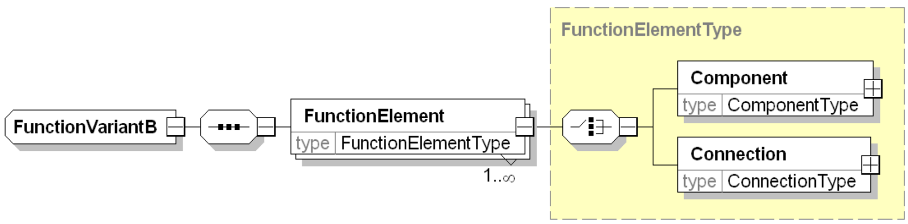

Figure 38 – Function variant using a parent function element

Finally, a namespace could be assigned to the schema to avoid interferences with other XML documents. In this thesis prototypical implementation, however, the namespace has been left out in favor of a better readability of the illustrated documents and diagrams.

The proposed schema implementation is considered as a basic solution to provide a description for function-oriented data. The format is designed with a focus on simplicity in order to reduce error probabilities and barriers for practical utilization. If necessary, this implementation can be modified or extended as needed to suit additional requirements. So, other solutions are possible as well and their suitability may differ, depending on particular use cases and system environments.

Figure 39 shows the complete structure of the final XML schema.

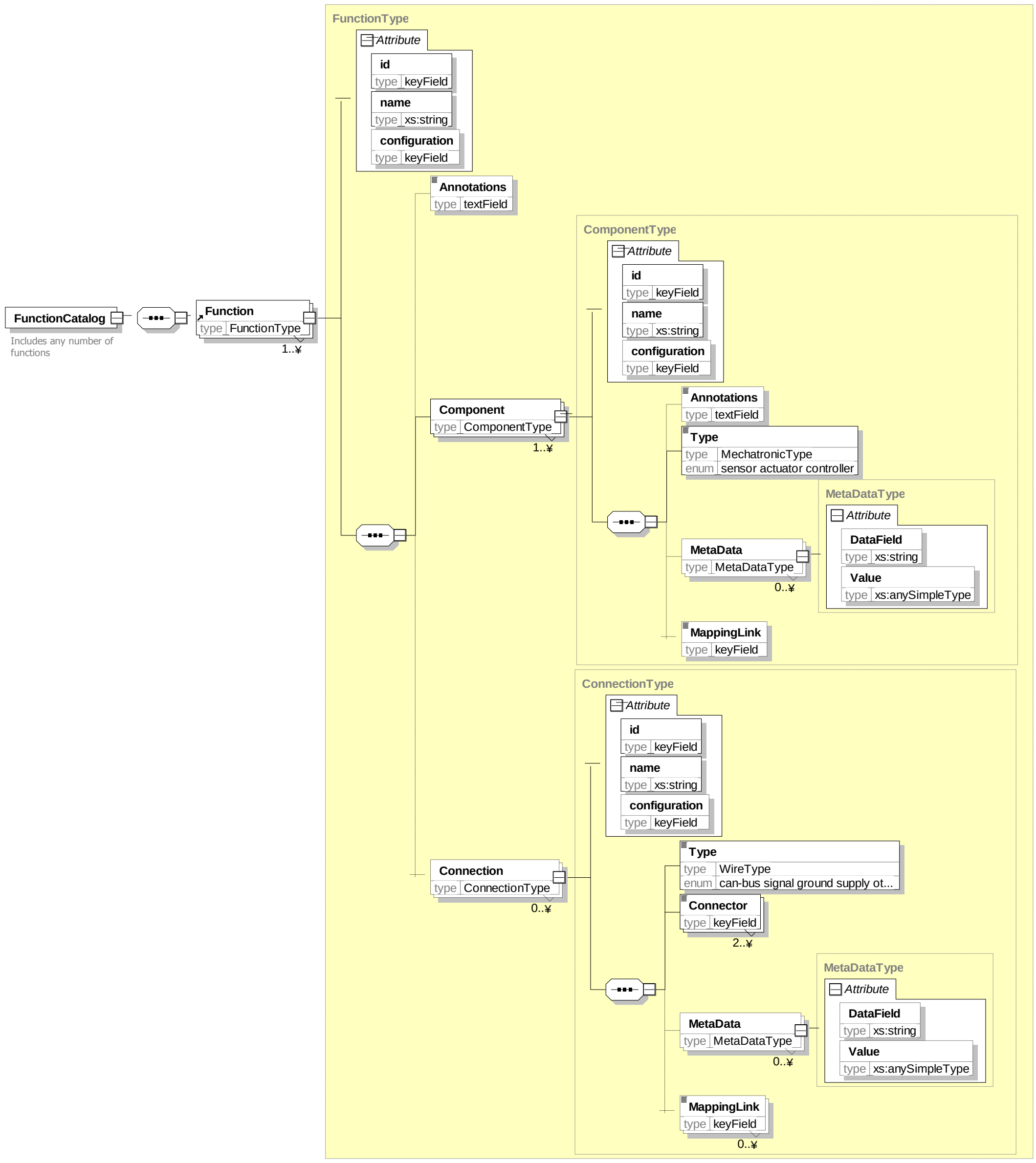

Figure 39 – Function-oriented XML schema diagram

Summarizing, the above XML solution contributes to prove the feasibility of the data integration concept. With the function data available in an exchangeable format, it is comparatively easy to exchange such data and provide XML interfaces to make the data usable across different tools. While this work focuses on the integration of function architectures with CAD data, the approach is not limited to this particular application. Instead, the proposed XML format enables an interdisciplinary utilization of function architecture data so that this data can also be provided to other domains of automotive development as well.

The complete XML code of the schema is to be found in Appendix A.2. In addition, Appendix A.3 shows an example of basic function architecture data that is converted into the XML format.

5.3 Comparison of Two Technical Data Integration Variants

In the previous sections, I have proposed theoretical prerequisites for a consistent mapping and an XML-based format for the description of function architecture data. Based upon these works, I present a technical concept for the integration process of function data with CAD data.

5.3.1 Integration Concept Main Steps

At a high level, main steps in the process of data integration and application can be identified as follows:

1 **Acquisition of the CAD data**
The CAD data is provided (e.g. exported from a data management system) using an appropriate volume of data that fulfills the needed requirements at consistent mapping.

2 **Acquisition of the Function Data**
The function data is provided (e.g. exported from a function architectures system and using the proposed XML-based meta-format) so that it can be integrated with the CAD data. Dependent on the format and system, it may be necessary to use a custom interface to fetch the data.

3 **Data Integration**
Both sets of data acquired in process steps 1 and 2 are integrated using an appropriate interface.

4 **Application of the merged Data**
The integrated data is ready to be used for function-oriented tasks within an appropriate target system that is capable to properly handle the function-oriented 3D data.

Technically, there are different possible ways to technically implement the above process steps. In the following, two different basic approaches are discussed which mostly differ in the implementation of the data integration interface.

5.3.2 Variant A: External Interface

This variant focuses on integrating the function architecture data (XML) into a CAD structure data file (PLM XML). Therefore, the CAD data is dually exported as a JT file and as a PLM XML file from the data management system. The function architecture data is saved as XML. In this variant, the integration interface is an external, custom data merging tool that integrates the function XML data into the PLM XML file using defined mapping rules and conditions. In this solution, the data with function-oriented information is added to an established standard file format (PLM XML) which is loadable in a common visualization system. Figure 40 illustrates this approach.

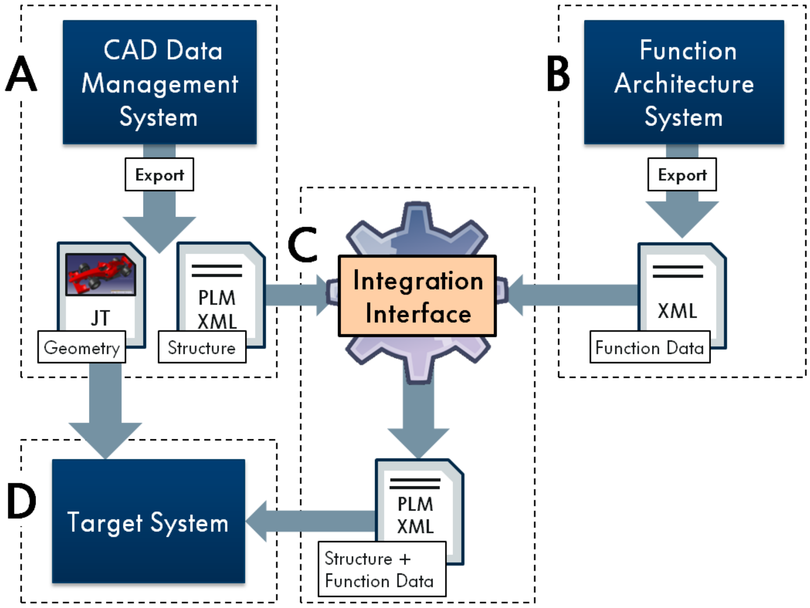

Figure 40 – Integration concept variant with function-oriented data merged into a PLM XML file

This approach requires a custom data mapping tool, respectively an integration interface, to include the function data into the PLM XML file format. This variant of implementation easily fits into existing and established standards and processes because function-oriented information are added to a standard file format and there is no need to make changes to a standard visualization system.

5.3.3 Variant 2: Internal Interface

The second implementation variant that is encouraged by the XML-based utilization of function data intends in leaving both data sets, the function data and the CAD data, in their prevailing file formats and separately loading them in the visualization system. Therefore, this variant proposes that the interface is implemented within the target system so that this system is able to import and process function-oriented data in the XML-format. A potential work flow might be that a user first loads the CAD data and then uses a specific integrated interface to additionally load the function-oriented information. Similar to the first variant, the interface needs to specify the rules for the data mapping. Figure 41 illustrates this approach.

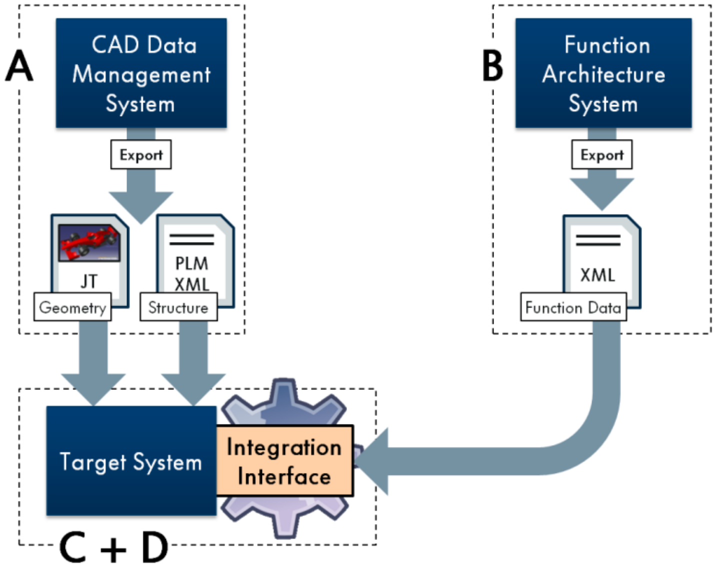

Figure 41 – Integration concept variant with separated function-oriented data

This variant is more of a modular approach that maintains a clear data separation and it might be an option if a 3D visualization system is used which does not support the PLM XML format. However, this approach requires changes to the target system in order to implement an interface to support the function-oriented XML data.

5.4 Prototypical Implementation

I have developed a prototypical implementation of the proposed data integration concept which complies with two primary goals. First, the prototype shall prove the feasibility of the integration concept within current automotive systems and processes. Second, it enables the development and exploration of novel function-oriented 3D methodologies in order to improve and streamline relevant tasks within a function-oriented development (see Chapter 6 and Chapter 7).

The prototypical implementation of the data integration concept has been developed using real CAD data of virtual vehicle prototypes to ensure a representative and realistic scenario. As proposed in the integration concept, the CAD data has been exported from a data management system using JT and PLM XML file formats. The geometric data has been divided into separate groups, including *vehicle chassis*, *electrical system* and *assembly components*, as shown in Figure 42. This grouping allows for easy selection and processing of exclusive parts of a particular group while leaving the other groups' content unaffected. For instance, this grouping allows visualization of the vehicle chassis in transparent colors, while particular components and/or the electrical system can be highlighted in solid colors so they can be clearly identified in the complete context of the vehicle.

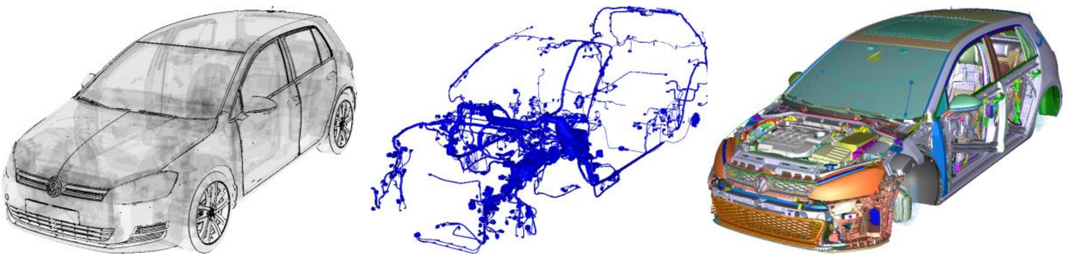

Figure 42 – CAD data clustering. From left to right: (a) vehicle chassis (exterior), (b) electrical system, (c) remaining assembly parts

The prototypical implementation uses function/system architecture data as it has been described in Section 3.4, including information regarding involved components, connections and respective types of such function elements. The functions which have been integrated for the prototype include vehicle functions described in Section 3.3, including park assist, light assist, start-stop automation and headlamp flasher. As it concerns the two implementation variants proposed in the previous section, variant A had been chosen for the prototype because it requires no changes to a widely-spread, heavily-used third-party system and because it enables seamless implementation into the existing processes which are present in the automotive company that supported this thesis work.

A custom external interface tool has been developed which enables an integration of function XML data into PLM XML files (see Appendix A.4). For the prototype, the required mapping links have been manually defined because the required identifiers are not yet implemented and maintained in the current processes and tools within the industrial work environment that supported this thesis.

The prototypical implementation has been done within an established 3D visualization system that is also used for regular DMU applications within the industrial working environment which hosted the work of this thesis [48]. This target system choice ensures a fair user-experience because the integrated data can be utilized in a tool that is well-known by the users. In addition, methods which were already available in the tool for geometric data operations (e.g. object manipulation, rotation, zoom, grouping, filters, different visualization variants, etc.) could be used as a basis for the function-oriented 3D methodologies.

Summarizing, this work allows the utilization and research of 3D CAD data which is enriched with function-oriented data within an established 3D visualization system. An in-detail investigation of the capabilities of the prototype and the elaboration of novel function-oriented 3D methodologies is proposed in the following Chapter 6. Notably, all contributions, contents and 3D artifacts shown in the subsequent chapters are based upon this prototype.

6 Novel Methodologies for Visualization, Analysis and Communication of Function-oriented Product Data

The research in this chapter focuses on the novel possibilities which are enabled by the implemented data integration concept proposed in the previous chapter. So, first, this work further verifies the applicability of the integration concept. Then, based upon the prototypical implementation, it focuses on development and research of novel, function-oriented 3D methodologies for interaction, visualization, analysis and communication of function-oriented data. In addition, based on these methods, novel features are proposed which can to be added to current and next generation 3D visualization systems in order to improve their capabilities to better utilize function-oriented data. Eventually, the proposed methodologies are applied and evaluated in different automotive use cases. Figure 43 highlights the focus of this work.

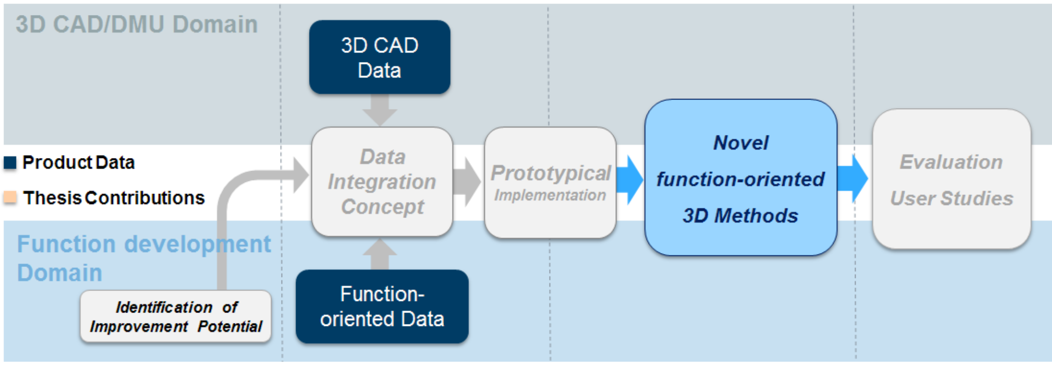

Figure 43 – In this chapter, novel function-oriented 3D methods are developed and investigated in context of different use cases within automotive PLM

6.1 Novel Methodologies for Interactive Spatial Visualization of Automotive Function Architectures

6.1.1 Function-oriented 3D Visualization

A *function-oriented 3D visualization* is a novel methodology that is enabled by the proposed data integration concept. Due to the prototypical implementation in an established visualization system, the basic rendering capabilities of this system can be exploited for the function-oriented 3D visualization. Figure 44 illustrates some examples to demonstrate different variants of visualization. Notably, these examples do not feature any type of function-oriented visualization yet.

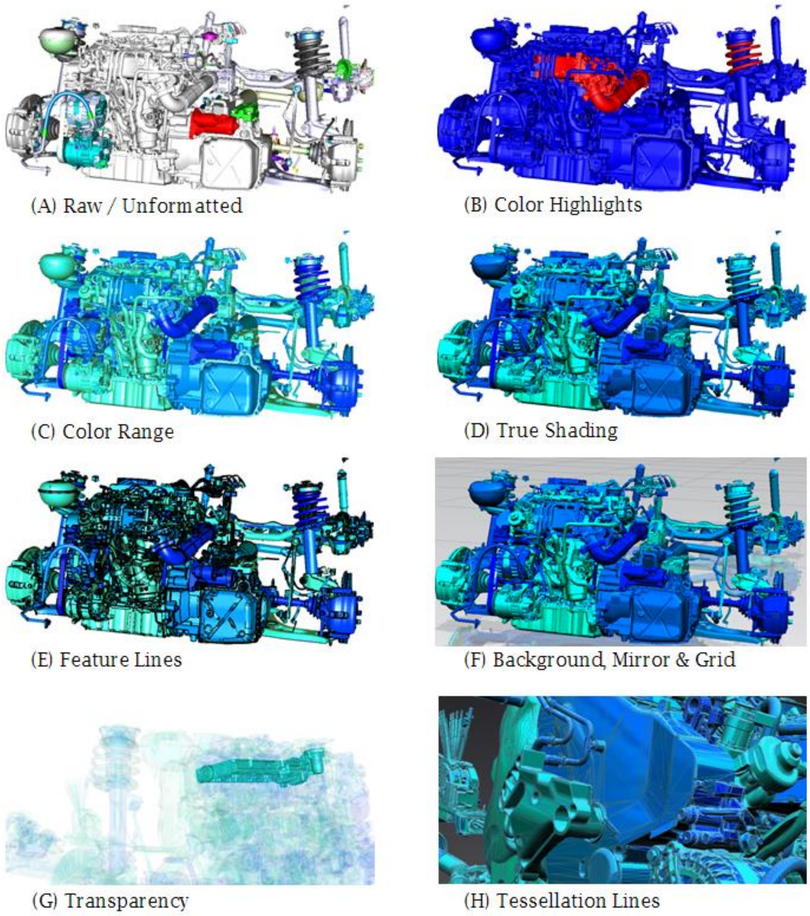

Figure 44 – Different exemplary variants of geometric data visualization which are supported by current CAD 3D visualization tools

A primary use case of the function-oriented 3D visualization methodology is highlighting of components and/or connections that are related to particular vehicle functions in order to solve the GFLT (see Chapter 3.5). Based upon the prototypical implementation, I have implemented this use case on the example of an automotive headlamp flasher function. Figure 45 illustrates a slightly simplified exemplary version of an architecture diagram of this headlamp flasher function.

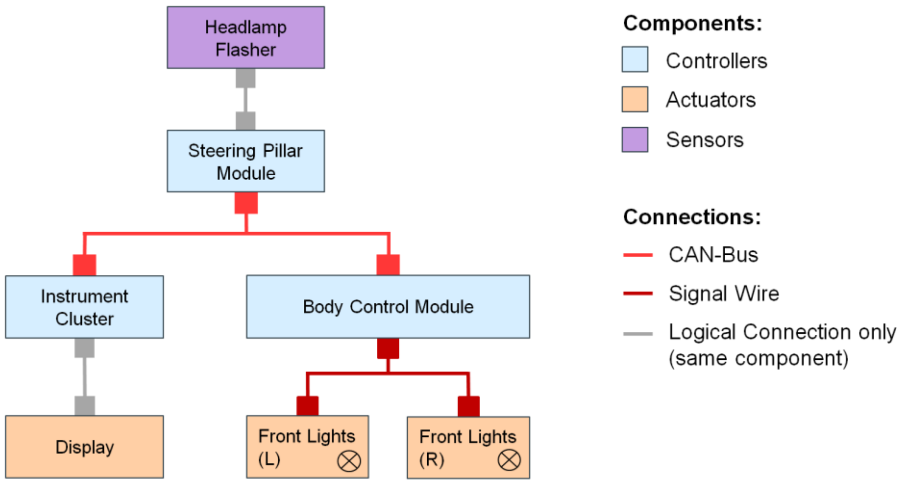

Figure 45 – Architecture diagram of a headlamp flasher function

In the prototypical implementation, the above architecture data has been integrated with 3D CAD data of a virtual vehicle prototype. This allows for a function-oriented 3D visualization as illustrated in Figure 46. This visualization highlights and colorizes all geometric parts of the vehicle that are related to the given function-architecture and clearly shows their spatial distribution and location in the particular vehicle assembly. Due to the color coding, connections between the components are traceable in the virtual vehicle and it is visible which geometric parts relate to which elements of the function architecture.

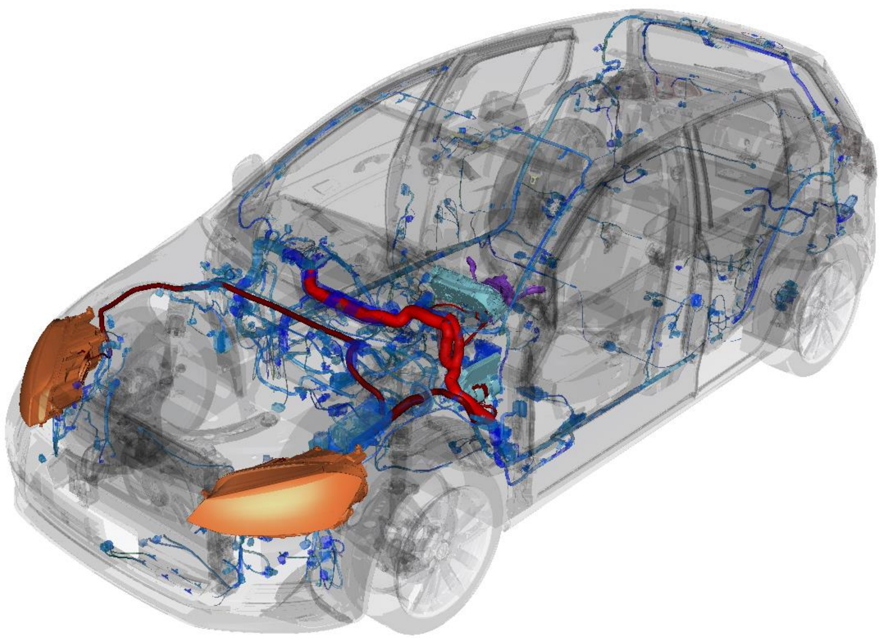

Figure 46 – A function-oriented 3D visualization of a headlamp flusher function in a virtual vehicle prototype that has been created with the novel methodology

Notably, the given example depicts a case that includes more function architecture components than could be possibly assigned to geometric parts. The headlamp flasher (sensor) and the steering pillar module (controller) are two separate blocks in the architecture diagram. However, in the actual vehicle assembly, both of these function components are implemented in the instrument cluster which is a single geometric part.

The above function-oriented visualization is one possible variant of visualization. Particular settings like color, transparency, perspective and orientation can be varied and customized to best fit the underlying use case. For instance, Figure 47 illustrates another visualization variant of the same scene but with higher transparency values for the chassis and activated true shading for a more sharpened look so that the wiring harness data is better recognizable.

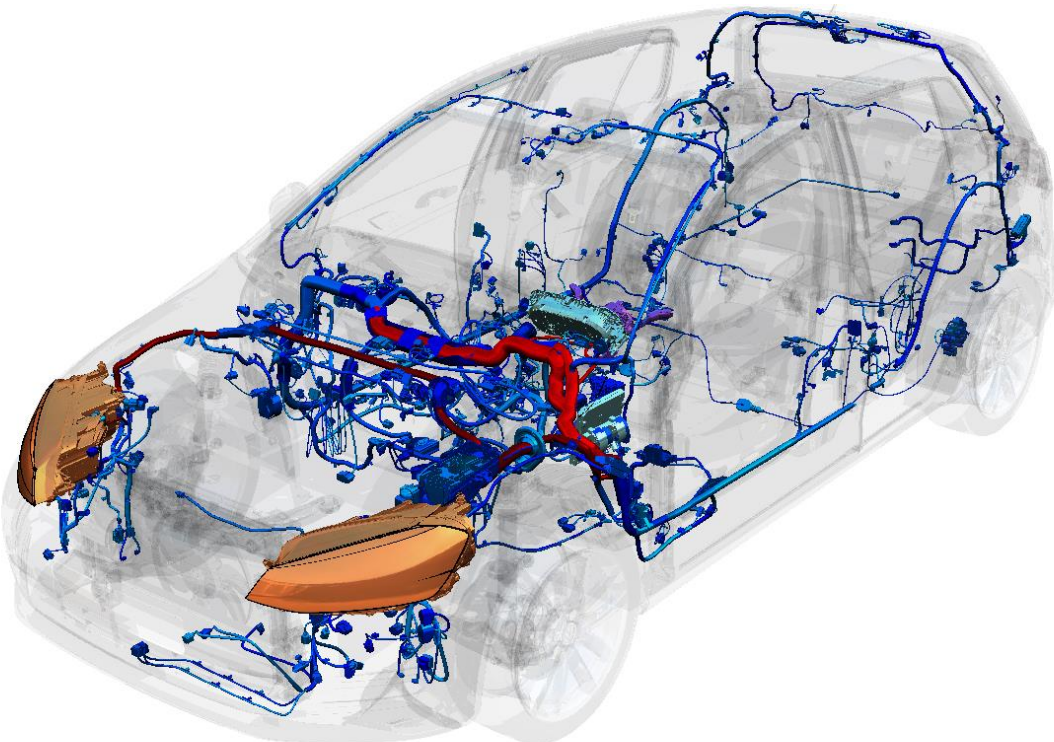

Figure 47 – Function-oriented visualization with the novel method and improved visibility of the function elements due to higher chassis transparency and more sharp wire shading

Due to the prototypical implementation within an established visualization system, many common operations for data manipulation and exploration are available to be used in the proposed methodology. For instance, such operations include zooming, rotation and filtering. Thus, it is possible to manipulate the visualization in real time and to approach the data from any point of view. For example, Figure 48 and Figure 49 show the same data as in Figure 47, but from different points of view.

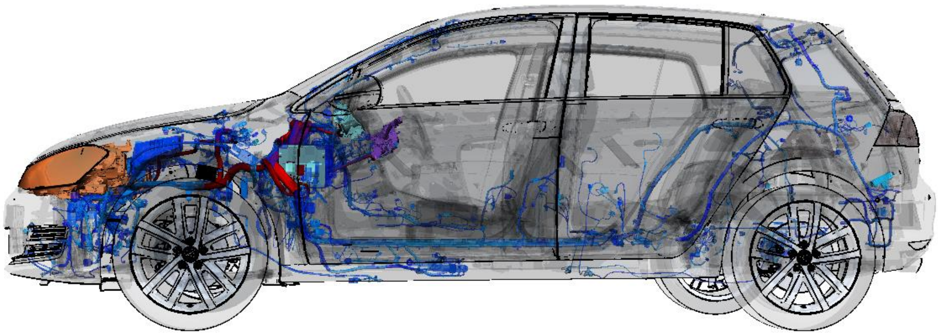

Figure 48 – Function-oriented visualization from cross-section view and with feature lines

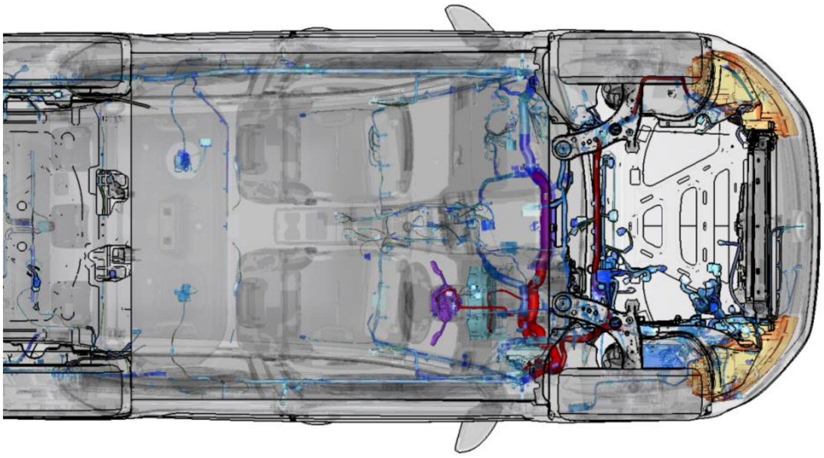

Figure 49 – Function-oriented visualization from top view

The proposed function-oriented 3D methodology also allows for simultaneous integration of multiple function architectures with a single set of geometric CAD data. Thus, for example, it enables spatial comparison of different vehicle function architectures in a specific car project which is a considerably powerful feature for function-oriented visualization. For instance, Figure 50 shows an example that highlights function components of three different vehicle functions in a car front end area from a top down view. From left to right, functions are *Park Assist*, *Start-Stop Automatic* and *Dynamic Light Assist*. Without this feature, a comparison task would usually involve significant manual efforts, for example, studying multiple documents and talking with multiple different engineers so that the task might take multiple hours. In contrast, with the function-oriented 3D method, the comparison can be performed within a few minutes and the visual presentation of information can improve the overall understandability of the data.

Figure 50 – The proposed methodology enables spatial comparison of multiple function architectures in specific vehicle projects. Left: park assist, middle: start stop, right: dynamic light assist

6.1.2 Function-oriented Data Analyses

The system-independent data structure used in the prototype (see Chapter 5.2) includes the description of function-oriented data and also additional custom data. The integration of this data with the geometric CAD data enables novel methods for data analyses which are explored in this section.

The 3D visualization system used for the prototype provides basic functionality to perform information-based automated analyses and tasks involving user-specific data which has been integrated with geometric data. This functionality is based on logical processing blocks:

- *Input data*
- *Processing rules*
- *Actions*

Input data defines the geometric parts that are to be affected by the analysis, including options such as: *visible parts*, *selected parts* and *all parts*. The *processing rules* define logical mathematical statements that need to be true for any actions to be triggered. The *actions* process the input data based on the specified processing rules and allow for operations like *show only*, *add to show*, *hide*, *select only*, *add to selection*, *deselect* and *change appearance*. Changes in color and appearance of parts can be separately defined for matched and unmatched results. For example, matched results can be colored red while unmatched results are rendered in transparent grey.

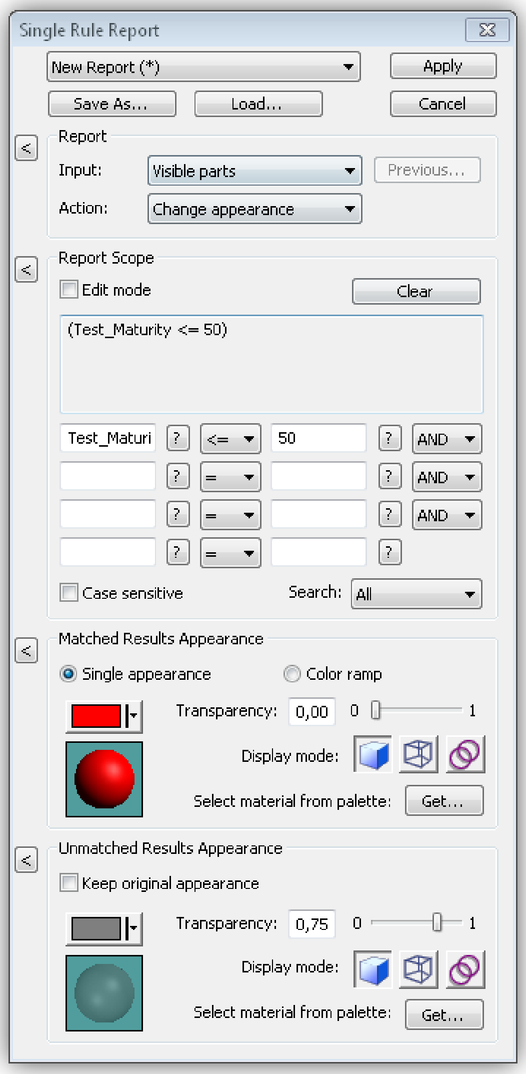

Figure 51 – An example of reporting functionality within a common visualization system

Figure 51 illustrates a dialog window in the used tool for setting up an analysis task with a single processing rule. In this example, geometric parts with a *test maturity* of 50 or less are colored in red, while all other parts are rendered in transparent grey. So, this automatic analysis highlights all vehicle parts that lack a specified test maturity. The test maturity is an example of custom data which has been added to the function-oriented data in order to provide advanced possibilities for data analyses. Another practical use case of this method is discussed and explored in Section 6.2.5. Notably, such kinds of analyses already provide beneficial new functionalities to support a function-oriented development. However, they are still limited regarding their capabilities of exploiting all potentials of the integrated function-oriented data. For instance, a more advanced use case could be:

"Show me all components related to the CAN bus of the headlamp flasher function"

However, this use case is not yet addressable with the previously discussed basic functionalities. For example, the input data is not able to process function-oriented information. Consequently, in the following it is explored which basic features should be made available for respective 3D tools in order to provide better support to process function-oriented information.

First, I suggest that actions can be performed which should be able to target *functions* as well as particular *function elements*. For *visualization* tasks, actions should be available to affect visual properties of the picked elements. Basic actions which are useful in multiple use cases are:

- *Select / Unselect*
- *Show / Show exclusively / Hide*
- *Set color / set transparency*

To streamline workflows, it should be possible to combine multiple actions in any order and to create presets. For example, a useful preset would be as follows:

- *Set color of all elements of a selected function to a specific color and make everything else transparent*

This preset is supposed to make all elements of a particular vehicle function easily recognizable within a virtual prototype. Such and similar types of use cases focus on the selection and visual highlight of specific subsets of function elements.

The use of *conditions* allows exclusive execution of actions (like *visualization* or *selection*) on a set of functions or function elements matching the particular conditions. For example, an engineer might be interesting in highlighting only those function elements which match a particular condition. This can especially help to streamline *search* tasks within the function data and which are very common tasks within function-oriented development. Thus, conditions are a strongly relevant feature to improve the capabilities of 3D visualization tools to better utilize function-oriented data.

For example, in case that the loaded function data includes additional custom data, the custom data could specify a *test maturity* for each controller. In an exemplary use case, the user wants to highlight all controllers in red color, that have a test maturity lower than 50. Thus, a condition needs to be specified which checks if the data field *test maturity* is lower than 50. Then, an action could be connected with the conditions, for example *selection* of geometric parts. As a result, all geometric parts are selected which math the condition. A second action could colorize the selected parts in red color. In pseudocode, the workflow of the tool would be as follows:

- *Select all parts matching condition [test maturity < 50]*
- *Set color of selected parts to red*

For this use case, a preset could be defined as follows:

- *Show me all function elements with insufficient test maturity*

Conditions should be made combinable so that multiple conditions can be linked with logical Boolean algebra disjunctions like *AND* and *OR* operators. Conditions should not be limited to numeric comparisons but could also include text comparisons and possibly more advanced types if needed.

The combination of actions and conditions allows to cover a wide range of use cases which can be phrased in pseudo-code like this:

- *For a given function, get all elements matching condition X and perform an action*
- *Get function(s) for a given element and perform an action*

More specific examples are:

- *Show me all functions, where currently selected components are involved*
- *Show me all components, which are involved in the same function, as the selected part*

Furthermore, many other examples are possible.

Without such methods, if an engineer is interested in the spatial position of geometric parts which match particular conditions, he would first need to separately gather the function-oriented information and custom data, which can be a cumbersome process on its own. Then, he would need to manually search the respective geometric parts within the 3D data, which would be a process that is significantly time-consuming and also prone to errors. Thus, the function-oriented 3D analyses explored in this section provide quite powerful new working tools for engineers which are not available without a consistent integration of function-oriented data with geometric 3D CAD data.

6.1.3 Interactive function-oriented Data Exploration

Based upon the function-oriented 3D methodology, the workflow is made possible that an engineer can use the geometric data or, rather, its visualization to select a particular geometric part in order to access related function-oriented data. This way, it is possible to easily and intuitively obtain function-oriented information related to particular geometry. This method is a reversed approach in comparison to the prior discussed methods and it helps to address different function-related questions, for example:

- **Functional relations**: To which function(s) does this geometric part belong to?
- **Type**: Is this geometric part a controller, sensor or actuator? Or, if it is a wire, is it a signal, grounding, etc., wire?
- **Network relations:** If the part is a wire, is it involved in any CAN-networks or other buses?

These are a few examples of questions that can be answered by the proposed methodology. Theoretically, *any* data that is available from the function architecture diagrams or obtained by other sources, and which has been input in the data integration, can be evaluated in related analyses and applications. Therefore, for example, the methodology can be used as a complement for digital mock-ups, enhancing geometric CAD data with function-oriented information. As a result, the digital mock-up is able to perform multiple new types of analyses and investigations related to a virtual product prototype, providing potentials for improving product quality and making decisions or finding potential issues at earlier stages of development. In addition, it can provide a new way to explore data with the potential to provide a more interesting learning and unser experience for the engineers.

Figure 52 illustrates a practical example implemented with the prototype in which a *Body Control Module* (BCM) is selected in the geometric data of a virtual vehicle prototype with the goal of obtaining related function-oriented information for this particular component. An exemplary workflow could be as follows:

- First, the user right-clicks on the body control module in the visualization system (red marked part in Figure 52). Subsequently, a window is displayed that shows function-oriented properties related to the clicked component.
- Next, a click on a *GetFunctions* field opens another window listing all vehicle functions in which the selected component is involved.
- Then the user chooses a particular function from this list and clicks on it. In this case he chooses the headlamp flasher function.
- Eventually, all related elements of the selected function are visually highlighted in the virtual prototype.

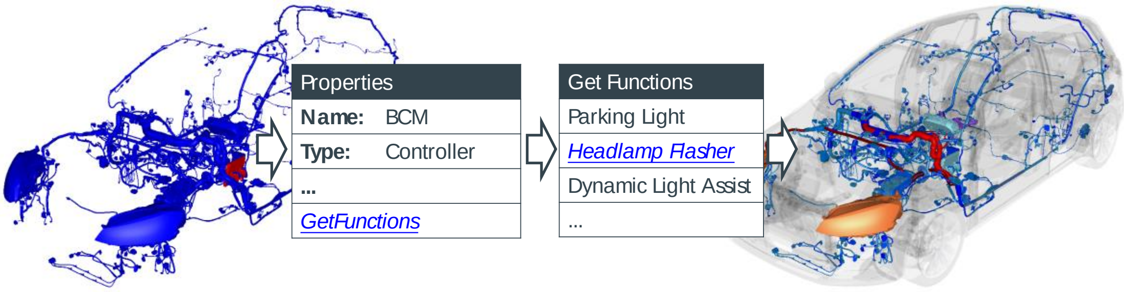

Figure 52 – The proposed methodology enables an interactive exploration of CAD data in order to obtain related, function-oriented information for selected geometric vehicle parts

I define this workflow with the concept sketch of a *function-oriented context menu*, shortly called context menu. This is a feature which can be added to 3D visualization systems in order to improve their capabilities of working with function-oriented information and make function-oriented workflows more efficient and intuitive. With this workflow, a user can quickly explore the geometric data and click on a geometric part of interest to select it. Then, the user can use a right-click (or a specific key) to expand a function-oriented context menu which is dependent to the selected part. Due to the integrated function-oriented data, the context menu knows all function-oriented properties of the selected part. In addition, the context menu is able to provide beneficial features such as:

- *Show/select all functions where the selected part is involved*
- *Show/select all other parts which are used in the same function(s) as this part*

More features could be enabled if concurrent selection of multiple parts is supported as a feature in the respective 3D tool. Then, for example, a user can select multiple geometric parts at once and the context menu could provide features such as:

- *Show function(s) where all selected parts are involved*
- *Highlight the connections between the selected parts for one or more specific function(s)*
- *Show all shared connections between parts across a specific range of functions*

This novel function-oriented 3D method provides a very fast way to retrieve function-oriented information by intuitively exploring geometric CAD data. It does not only provide beneficial new workflows for automotive development, but also for learning purposes and the service and maintenance domain. For example, a service technician might know that a particular component is defective. With the novel methodology, he can look up this component in the visualization, select it, and thus quickly obtain related function-oriented information. As a result, he gets to know which functions may be potentially malfunctioning due to the defective component (also see Section 6.2.2.).

6.1.4 Data Management

In many of current automotive CAD and DMU-oriented data management and visualization systems, navigation hierarchies of product data do not yet feature function-oriented structures. For example, such hierarchies are usually derived by assembly zones and/or other given structures of the CAD data or data management system. Such hierarchies mostly represent the classic granularity in the automotive industry where components are arranged by technical groups (e.g. cockpit, spotlights, chassis, etc.). Consequently, such hierarchies do not particularly support function-oriented data navigation and exploration. Moreover, as illustrated in Figure 53, hierarchy trees can be considerably cryptic which, for example, can be caused by technical naming conventions and encodings. This makes such kind of data structures difficult to understand, maintain und use for humans.

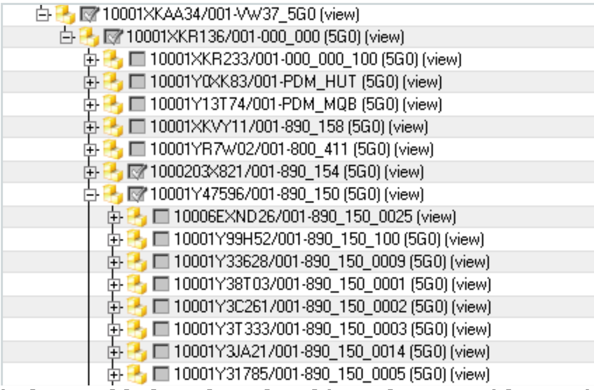

Figure 53 – Hypothetical assembly-based product hierarchy tree with cryptical node descriptions

Function-oriented applications are growing increasingly important. Thus, the approach of this thesis suggests to consequently provide function-oriented hierarchies in respective data management systems in order to streamline data handling for function-related analyses, and to generally improve the user interface to access and work with such data. Basically, such hierarchies are supposed to be human-readable and to support an intuitive and fast analysis of function-oriented data.

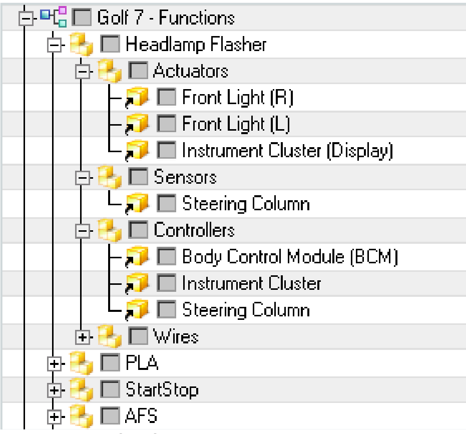

Figure 54 – Example of a function-oriented product hierarchy tree

Figure 54 shows a basic example of a function-oriented data structure that has been used in the prototype and which can support engineers with better capabilities to manage function-oriented data structures. To give an example of a more sophisticated implementation to improve function-oriented data management, I have developed a concept sketch which I call *function explorer* and which is illustrated in Figure 55.

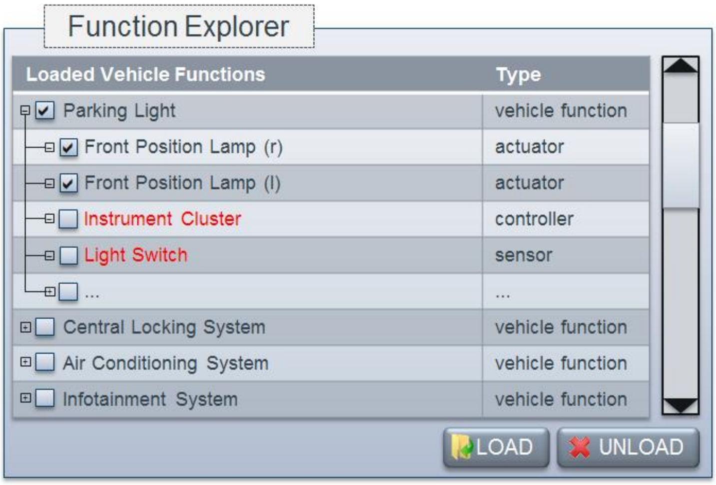

Figure 55 – The function explorer concept sketch shows a possible approach to improve function-oriented data management

Basically, the function explorer is a hierarchical, function-oriented navigation window which, for example, could be made a part of an add-on to enhance a 3D visualization software with function-oriented features. In comparison to the function-oriented hierarchy shown in Figure 54, the function explorer provides additional information and functionalities. For example, the function explorer allows for separately loading and unloading function data which could be provided in the proposed XML format. After loading the data, it is displayed in a hierarchical tree structure. It is suggested that this structure lists at least the loaded function names, their elements and the elements types. Moreover, if needed, different kinds of additional data could be made accessible as well.

In addition, the function explorer should provide information on which function elements are properly mapped to geometric parts of the CAD data. In case that a proper mapping is not possible, for example due to missing key fields, elements should be marked in red so that a user is notified about possible issues. This feature helps to quickly detect insufficient quality of data and to identify potential gaps in the related development processes. Moreover, the function explorer should enable different types of interaction with the loaded function data. For example, checkboxes can be used to quickly mark and select functions and/or their particular elements. So, the function explorer or a comparable tool is supposed to streamline the workflows for function-oriented data analysis as described in the previous sections.

Summarizing, this design sketch provides a first, basic suggestion to improve function-oriented data management. While this concept can and should be extended as needed for particular use cases, I recommend to consider at least the following minimum features:

- Loading/Unloading function data
- Provide information on loaded functions and their properties
- Provide information on data quality and mapping issues
- Selection of particular functions and/or function elements

6.1.5 3D Rasterization and Automation

A general and primary advantage of using computer-aided methods is their capability for automation. In many industrial use cases, if automation is possible, it can significantly improve speed and output quality of workflows. In this section, based on the novel function-oriented 3D methodologies, I present a 3D rasterization approach which allows to perform multiple, important function-oriented tasks in an automated way with reasonable performance. This approach overlays an integrated function-oriented 3D geometry with a number of N rectangular volumes (cells) and aims at determining volumes which are intersected by particular elements of the function-oriented 3D data. For the intersected elements, related information can be retrieved in order to provide new options for function-oriented data analysis.

In order to demonstrate both the motivation for this approach and its potential, I first present two examples as follows. For both of the examples the geometry has been loaded in the freely available 3D graphics software Blender (version 2.74) and the rasterization approach has been implemented using the Blender Python API as it will be described later [182]. In the first example, function-oriented geometric data of a vehicle wiring harness is used. This data is overlaid with a Cartesian regular grid. The result is shown in Figure 56.

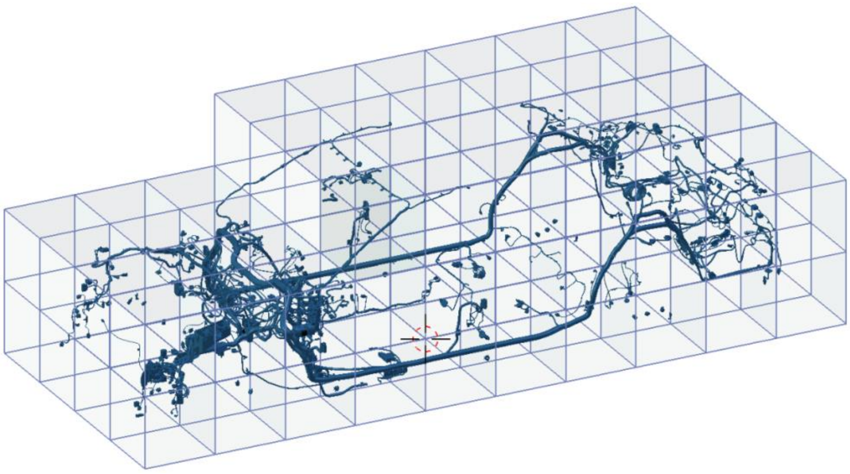

Figure 56 – A symmetrical grid is overlaid to the geometry to provide a basis for automated analyses that aims at detecting intersected grid cells

The above data structure provides a technical groundwork for precise, automated analyses of function-oriented data. For example, an engineer might be interested in the total distribution of wires which are related to a particular function in the vehicle. To answer this question, I calculate kind of a wiring density as

$$d_w = \frac{N_w}{N_c}$$

where N_w is the number of cells intersected by the function-related wires, and N_c is the number of cells touched by any geometry of the car. The calculated value d_w multiplied with 100 provides the overall distribution percentage of the wiring harness of the particular function within the vehicle. This result, for example, can be used to compare wiring distributions of different function architectures in order to save costs by reducing wire lengths. The accuracy of this method depends on the resolution of the grid so that a higher resolution with more cells of smaller sizes is able to provide a higher accuracy.

To discuss a second example which represents another type of use case, the same geometric harness data is overlaid with a number of rectangular volumes of manually-specified sizes as shown in Figure 57. In this example, the colored volumes represent crash-relevant zones in the vehicle with different deformation risks in crash situations. The *green* colored zones correspond with low risk, *yellow* means a medium risk and *red* means a high risk.

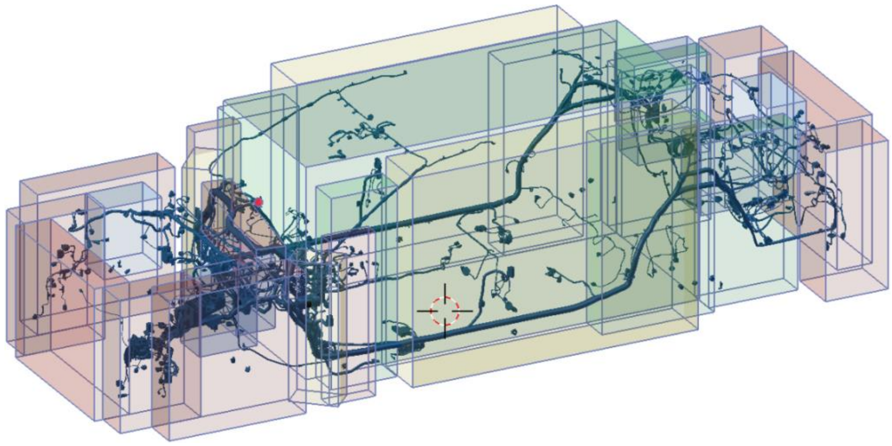

Figure 57 – Geometric vehicle data is overlaid with volumes representing crash-relevant zones where colors indicate deformation risks for the area

This uses case raises the question of how many elements of a particular vehicle function are distributed in particular crash-relevant areas. For instance, this information can be used to evaluate and compare function architectures in terms of safety and crash-robustness. Other similar and typical examples of using such user-specific areas in multiple automotive use cases include *temperature zones*, *vibration stress* areas, *mechanical shock* areas or areas *prone to dust and splash water*. In all of these exemplary use cases, the approach can be applied analogically as demonstrated in the example shown in Figure 57. Notably, the question of this use case (intersection of function elements with particular areas) can be answered by manually investigating the geometry within a 3D visualization tool or with an automated data analysis.

As it regards manual data analysis, users can explore and analyze such data as shown in Figure 57 in a suitable 3D application such as Blender in order to identify particular vehicle parts which are located in respective zones. Consequently, in the following Chapter 7, a user study is presented which evaluates such manual analysis of 3D data for this use case. Moreover, this approach is convenient to be implemented in an automated way as well. Thus, we have prototypically implemented an automated method in Blender using the Python API to solve the task of detecting geometry intersections with both grid cells and specific areas within a three-dimensional Euclidean space.

In the following, I present the prototypical automated implementation of this approach. This implementation first creates a *search volume* based upon a given input geometry in order to define the overall grid boundaries. Typically, the input geometry could be the geometry that is to be analyzed or a simplified approximation of it. In the second step, our method creates a *grid rasterization* based on the search volume. In the third step, a *search geometry* can be passed. Considering the above examples, this search geometry would be the function-oriented 3D data that is to be analyzed. Finally, in the fourth step, a subset grid (*search volume raster*) is generated that exclusively contains the grid cells that intersect with the search geometry. Figure 58 illustrates this method and its particular steps. In the following, I describe the particular algorithms which we have developed for the implementation.

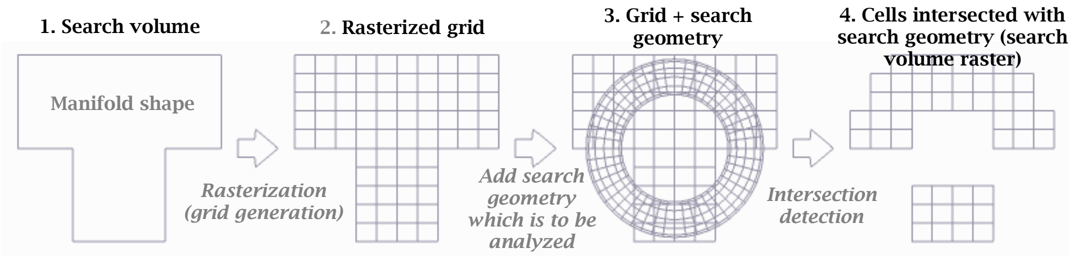

Figure 58 – Our implementation creates a rasterization (3D grid) from a given manifold geometric shape. Then, the algorithm detects the grid cells with intersections with the analyzed geometry

First, I present the *rasterization algorithm* (Algorithm 1) which creates a 3D grid of variable resolution from a given input geometry. Considering the illustration shown in Figure 58, this algorithm covers the first two steps which include generation of both the search volume and the rasterized grid. As it regards constraints, the input geometry must be static, manifold, unscaled and its coordinate origin must be within the center of its bounding box. As input, next to the input geometry matching above conditions, the algorithm expects the numbers of grid cells for each axis x, y and z. In addition, to consider use cases such as shown in Figure 57, the algorithm can optionally take as input a set of geometric regions which can include data sets of user-specified properties. In the following processing, these properties can be used to exclusively detect parts of the search geometry that intersect particular regions, or only those regions with specific properties. For usage of regions, the same constraints apply as for the input geometry. In addition, regions must not be cascaded, they must not have any intersections with each other and they must not be smaller than the grid cells. Algorithm 1 is described in pseudo-code as follows.

Algorithm 1: Rasterization (G, N_{xyz}, R)

Input: G = geometry, N_{xyz} = cell count in x,y and z dimension,
R = set of regions (optional)

Output: 3D grid rasterization

calculate raster cell size based on bounding box of G and N_{xyz}
calculate 1st cell position (use corner position of and within G)
create 1st cell and put it on correct location an rotation
save cell radius
for each axis x, y, z
    copy 1st cell to create full raster (3D grid)
endfor
set origin point of raster to bounding box center

```
for each grid cell in raster
    CC = get center point of cell
    check if CC is within G by sending out 3 rays from center point
                                                    (one for each axis x,y,z)
    if number of intersected faces is odd
        the point is within geometry
    else
        remove iterated grid cell
    endif
endfor
recalculate bounding box and set origin point of raster to bounding box center

if regions count > 0   // regions are optional
    Build k-d tree T which nodes include the grid cell positions
    Balance k-d tree
    V = Create hash table to store visited cells
    for each region in R   // iterate regions
        SR = calculate search radius
             (= region arbitrary boundary corner point - region center + cell radius)
        for each adjacent cell in find range of region center + SR
            if cell is not in V (visited cells)
                do nothing
            else if cell center is within region
                add cell to V (visited cells)
                P = get specific property of region
                save P of intersected region within cell
            endif
        endfor
    endfor
endif
return raster
```

In particular, Algorithm 1 first calculates the size of the grid cells based on the axes sizes of the input geometry bounding box and on the specified cell counts. Then, an initial start position is determined and the first cell is created with proper position and rotation according to the input geometry. A cell radius is calculated and saved for later reuse. Then, the cell is copied along all 3 axes in order to create a full 3D rectangular grid that matches the input geometry bounding box. Next, the coordinate origin is updated. Then, all cells are iterated and it is detected if they are within the actual input geometry or not. In the latter case, they are removed so that the final grid is a close approximation of the input geometry shape.

In addition, the algorithm can optionally take a list of regions. To process the regions, the algorithm creates and balances a *k*-d tree which nodes hold the positions of all grid cells. The *k*-d tree is used to improve algorithm performance based on the concept of reducing processing to adjacent cells, relative to the region position and its geometric spread. This allows for the improvement measure that not all grid cells need to be visited each time but only those with a high probability of being surrounded by or having intersections with the given geometry. In contrast to an Octree-based implementation, the *k*-d tree guarantees

a search complexity of *O*(kn) and allows for non-cubic grid cells which provide a more congruent and efficient solution to depict the geometry.
As follows, the algorithm iterates the given regions and calculates an appropriate search radius for each region as distance between an arbitrary region boundary corner point and the region center position and the cell radius which has been calculated for the raster object. Then, considering the region position and search radius, the *k*-d tree is used to calculate a set of adjacent grid cells which are iterated. Those cells which not have already been visited and which are inside the region will be enriched with the custom property data of the intersected region. In order to detect cells which are inside the region, the same intersection method is used as for the rasterization step. Finally, the algorithm returns the generated raster. Some examples on input and output generated with this algorithm are shown in Figure 59 as follows.

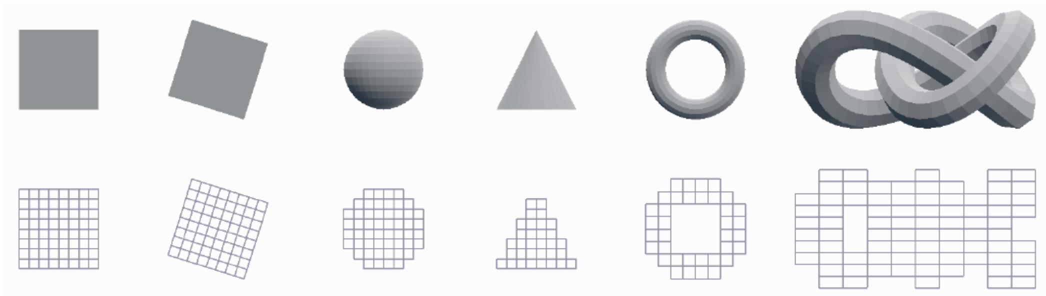

Figure 59 – Examples of rasterizations created with Algorithm 1 including input and output

Runtime performance of Algorithm 1 is *O*(N) for the creation of the rasterization part while *N* is defined by the total number of grid cells. If regions are used with the algorithm, total runtime performance can get close to *O*(N^2). So, the operation is relatively expensive because each cell needs to be passed through at least once and thus also significantly depends on the grid resolution. However, the grid only needs to be generated once and thus can be re-used afterwards.

As follows, I present the *search geometry raster detection* algorithm (Algorithm 2) which addresses step 3 and 4 of the rasterization method (see Figure 58). This algorithm takes as input a rasterized grid that has been generated with Algorithm 1 and a search geometry which needs to be a set of geometric objects within the same coordinate space. This covers the third step of our method as illustrated in Figure 58. For example, the search geometry can be a set of function-oriented 3D data. Based upon this input, Algorithm 2 generates a *search volume raster* which includes a subset of the rasterized grid, which only includes grid cells which do intersect with the search geometry to solve the fourth step. Algorithm 2 is described in pseudo-code as follows.

Algorithm 2: Search Geometry Raster Detection (*R*, *G*)

Input: *R* = raster object, *G* = list of geometries (*search geometry*)
Output: *OG*: output geometry (Search Volume Raster)

Build *k*-d tree with nodes holding all cell positions of *R* (the tree could also be re-used from previous algorithm)
Balance *k*-d tree
OG = Create *output geometry* as new empty geometry
CR = get cell radius from raster object *R*
V = Create hash table to store visited cells
for each geometry *g* in geometries *G*
    SR = Calculate *search radius*
        (= *g* arbitrary boundary corner point – *g* position + CR)
    for each adjacent cell in find range (based on *SR* and geometry center position)
        If cell not in *V* (visited)
            C = cell location
            NP = calculate nearest point of *g*, starting from *C*
            If *NP* inside cell **or** *C* inside geometry
                add cell to *OG*
                also copy all data, geometry, hash entry, position, and properties
            endif
        endif
    endfor
endfor
    update bounding box and center point of *OG*
return *OG*

Algorithm 2 first creates and balances a *k*-d tree which stores grid cell positions of the raster. Notably, the tree generated with Algorithm 1 could be reused as an optimization measure. Next, an empty geometry object is created which is supposed to store the cells of the output geometry which is to be calculated. Then, the cell radius from the given raster object is retrieved (this radius had been stored in the raster object data that has been created with Algorithm 1). Again, a hash table is used to store those cells which have already been visited within the following iteration. As follows, all objects of the input geometry are iterated. For each iterated object, a search radius is calculated similar to Algorithm 1 as distance between an arbitrary boundary corner point of the geometry object and its' center position plus the cell radius. Then, adjacent cells are iterated with the same approach as used in Algorithm 1. For each cell which has not already been visited, its location is stored. Then, starting from the cell origin, the nearest point *NP* of the given search geometry is calculated. Afterwards, it is determined if this point is within the grid cell. However, this intersection is not sufficient yet because in case a grid cell is fully surrounded by the search geometry, NP would not be within the cell even though the cell would be within the geometry. Thus, for manifold objects it is additionally calculated if the cell origin is within the geometry as well. Notably, other approaches would be possible if it can be assured that the search geometry is manifold as well (which could not be assured for the function-oriented 3D data that has been used within this research work). Then, the full data of the identified cell is copied into the final output geometry, including geometry, hash-table entry and all additional properties. Finally, the bounding box and center point are updated for the output geometry, respectively the search volume raster.

Runtime performance can get close to *O*(N³) at worst. N is defined by the product of the analyzed geometric objects, its adjacent grid cells and its common intersection within the search radius. The worst case would apply if there are at least as much particular search objects within the search geometry as grid cell while each search object takes up the space of a grid cell. Usually, however, runtime is much faster than O(N³) because intersection detection only applies if grid cells do not have already been passed through so that actual runtime is quite fast in most cases.

We have conducted some performance evaluations for both functions with a Mac OS system with Intel Core i7 processor and 16GB Ram. For the tests, Blender has been used with a Python interpreter, a single CPU process and a consistent runtime environment.

For the rasterization function, two cubes have been used as test hull geometry which then has been rasterized in different resolutions. The cubes have been used for the first 9 test cases with 3 different grid resolutions. In addition, different region counts have been used, including 0, 1 and 2. For the last 2 test cases, the same geometry has been used as in the user study which is presented in Chapter 7.2. The test case setup is based on the worst case so that the whole grid was always fully covered by one or two regions.

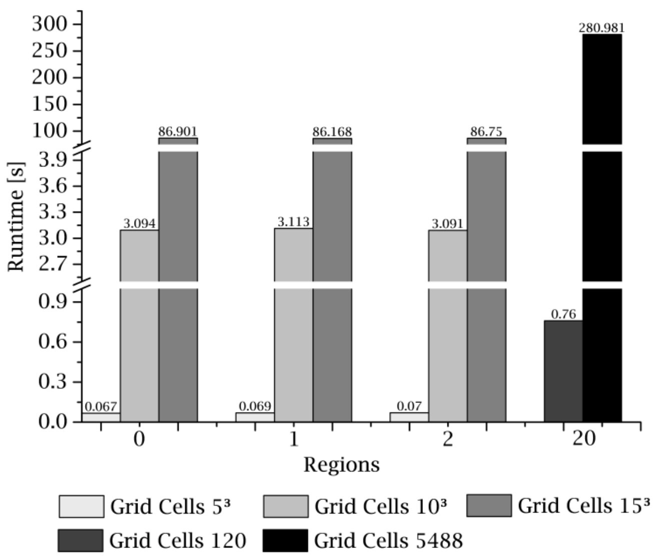

Figure 60 – Rasterization function runtime depending on number of grid cells and regions

Figure 60 shows the runtime evaluation results. It is notable that runtime increases with grid resolution. This is expected because all cells have to be processed at least once (including those not within the hull). The results show that the algorithm is very fast for smaller resolutions. Notably, however, the count of regions does not seem to have a significant influence on the runtime behavior. This can be explained because it is always checked if a grid cell has already been processed or not. Interestingly, the runtime is nearly equal between test cases with and without regions. Chances are that this result might be related to internal caching techniques of Blender. Notably, this interpreter-based implementation likely is not as performant as an implementation within a compilable programming language such as C or C++ which is expected to provide a considerably better performance, for example due to better possibilities for optimization measures and memory management.

Figure 61 shows test results of performance measurements for search geometry raster dectection algorithm.

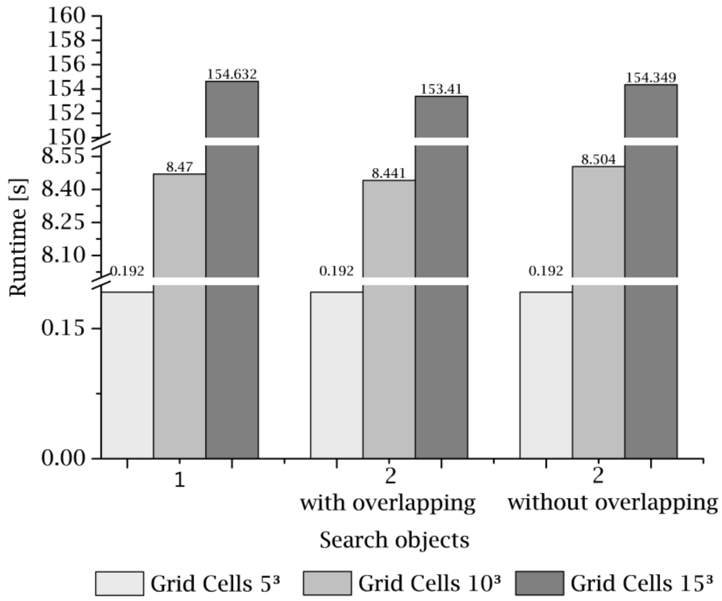

Figure 61 – Search geometry raster detection function runtime depending on number of grid cells

For this test setup, the geometric regions of the study from Chapter 7.2 have been used as search geometry. In total, 9 test cases have been performed, including three different grid resolutions with either 1 search object, 2 search objects, with and without overlapping. These test cases represent the times which are needed by the algorithm to either search 512, 1000 or 3375 grid cells for a given number of search objects. Similar to the previous test setup, runtime increases with the number of grid cells. Interestingly, runtime is sometimes lower for two search objects which again might be caused by internal caching techniques of Blender. Positively, it can be noted that runtime is lower for overlapping objects which shows that related intersected grid cells are only searched once for these objects. It can also be noted that overall runtime is higher than for the rasterization function, which can be explained by the additional computations such as intersection detection.

We have also evaluated the clustering function for a regular case with a test setup that uses real-world function-oriented geometry as search geometry. This function-oriented data features wiring harness geometry related to 3 different vehicle functions. This search geometry data involves 83 objects for the first function (F1), 71 objects for the second function (F2) and 133 objects for the third function (F3). These 3 functions have been rasterized with 2 different grid resolutions as shown in Figure 62.

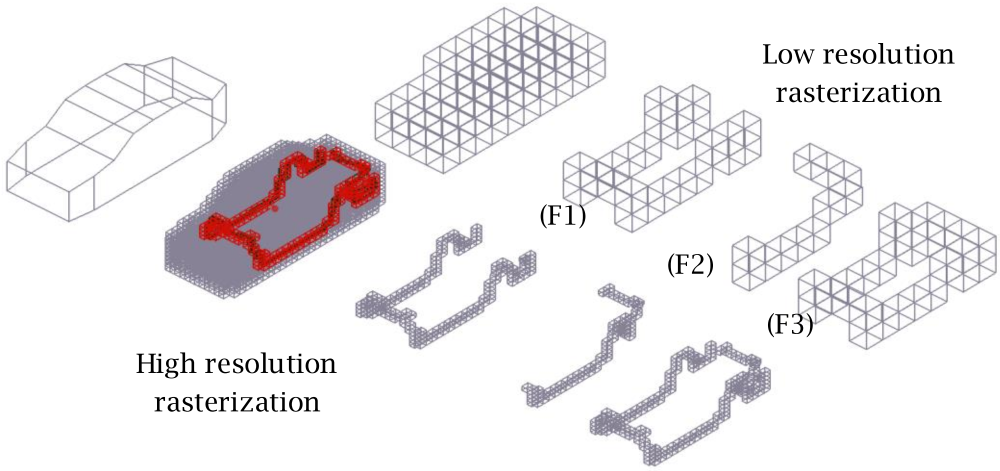

Figure 62 – Rasterization in two resolutions of function-oriented data (wiring harness) related to 3 different vehicle functions

The test results are summarized in Figure 63 show a very fast runtime performance for this real-world use case.

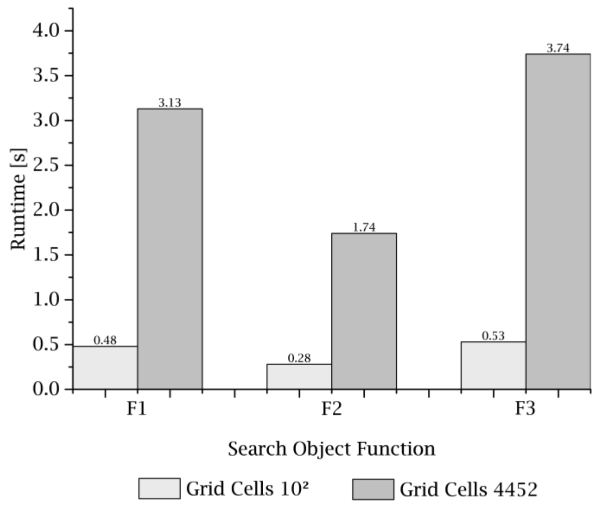

Figure 63 – Clustering function runtime evaluation with real function-oriented data

Summarizing, the results of the prototypical implementation are promising because the automated method is robust and able to rasterize arbitrary manifold forms and to identify a search volume raster for any given number of geometric objects independently from its topology. Moreover, the prototypical implementation in Blender has provided accurate results with reasonable runtime performance.

6.2 Benefits of the 3D Methodology in Automotive Use Cases

This section investigates use cases across different automotive domains where 3D function-oriented methods can be of beneficial assistance.

6.2.1 Generic Users and PLM Overview

The function-oriented 3D methodology proposed in this thesis can be useful for a significant number of stakeholders which are involved in function-oriented PLM. For example, these stakeholders include engineers, testers and service technicians who are involved in the development, verification and validation of vehicle functions and their related components. Other potential stakeholders include but are not limited to:

- Project and concept lead / control
- Technical departments of system and function supervisors
- Departments related to localization of interference source and sinks
- Departments for vehicle type approval
- Designers of system descriptions (system supervisors)
- Customer Service departments (e.g. maintenance, creation of technical documents)
- Functional safety departments (analysis and evaluation of errors and failure probabilities for ASIL-level classification)
- Wiring harness development, evaluation of options for wiring distribution in combination with safety-relevant functions
- Prototyping departments

6.2.2 Maintenance and Service

Naturally, the rising electric/electronic complexity of modern cars does not only pose a challenge for automotive development but also for the customer service. Consequently, this section discusses a typical use case in this area and highlights potentials of the function-oriented 3D methods to support this use case.

In the field of automotive service, when a malfunction is detected, it is a typical task for the service personal to identify, locate and repair or exchange the defective part(s). A function-oriented 3D-visualization can assist in this task by enabling a quick spatial determination of relevant assembly positions of particular components and wires. Therefore, the tracking time of specific components can be decreased, especially when considering that assembly positions of components can be different, depending on particular vehicle models and/or configurations. In addition, the visualization can be used in the training of new service technicians or to support the education after the introduction of new vehicle models. Furthermore, when a defective component is identified, the proposed methodology of interactive data exploration allows conclusions about other functions which may be potentially malfunctioning because they rely on the identified component.

For example, in both areas, development and validation of vehicle functions as well as repair and maintenance jobs, it is a common task to trace error causes based upon given failure symptoms. Such failures can be caused by line break (i.e. due to crash, fatigue, wear, etc.) and/or short-circuit (mass or 12V). In the following exemplary use case, an issue is assumed which is in the *power supply* of a *Steering Column Module*. This is a generic cause of error with a relatively high probability. In most current cars, power supply units are crucial components as they provide all systems with the necessary electrical energy. Usually, the electrical energy is created by a generator in the engine bay, temporally saved in the battery, and then distributed to the electronic components via energy dividers and/or fuse boxes.

This use case provides a fairly simple example of how the proposed 3D methods can assist testers and service technicians in the tracking of function-related wires. The architecture diagram in Figure 64 (left) provides information about the involved components and connections. The function-oriented 3D visualization in Figure 64 (right) complements this diagram by providing information about the actual location and distribution of these components and connections in the physical vehicle assembly. With the 3D method, the route of the power supply wire within the vehicle assembly is clearly visible, starting from the generator/battery (E-Box) over the load divider to the steering column module, as it also could be approximately expected by the assembly locations of the involved components. In this example, the topology of the diagram is comparatively closely related to the actual assembly topology. Notably, this is not necessarily true in practice for all cases.

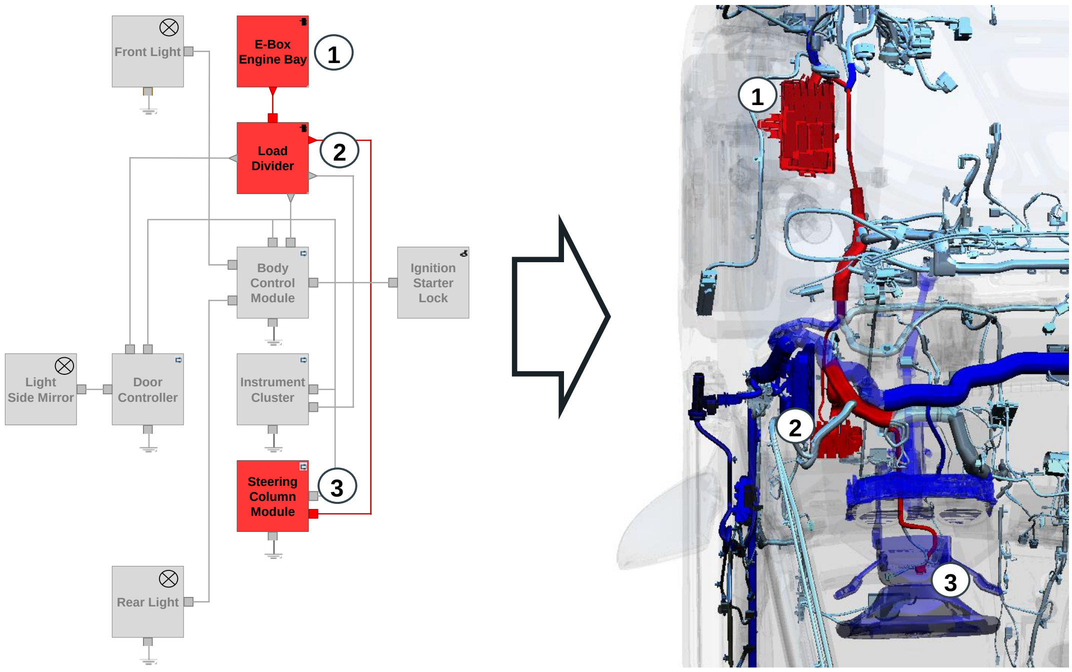

Figure 64 – On the left, all components related to the power supply of the steering column module are highlighted in the architecture diagram. On the right, the function-oriented 3D visualization shows accurate positions of these components and wires in the actual vehicle assembly

Another potential cause of error can be an issue with the *grounding connection* of the steering column module. Grounding wires connect function components to the vehicle chassis to enable a return of the electric current. This use case differs from the previous one in a matter that it strongly makes the limits of function architecture diagrams apparent, because, in this case, the diagram does not provide *any* information on the distribution of the grounding connection at all (see Figure 65, left). Moreover, the location of grounding bolts can be very difficult to detect in practice because they can be hidden behind caps and carpets. Dependent on vehicle platforms, design decisions and needs made by the chassis fabrication, the bolts can be located in different and unexpected positions.

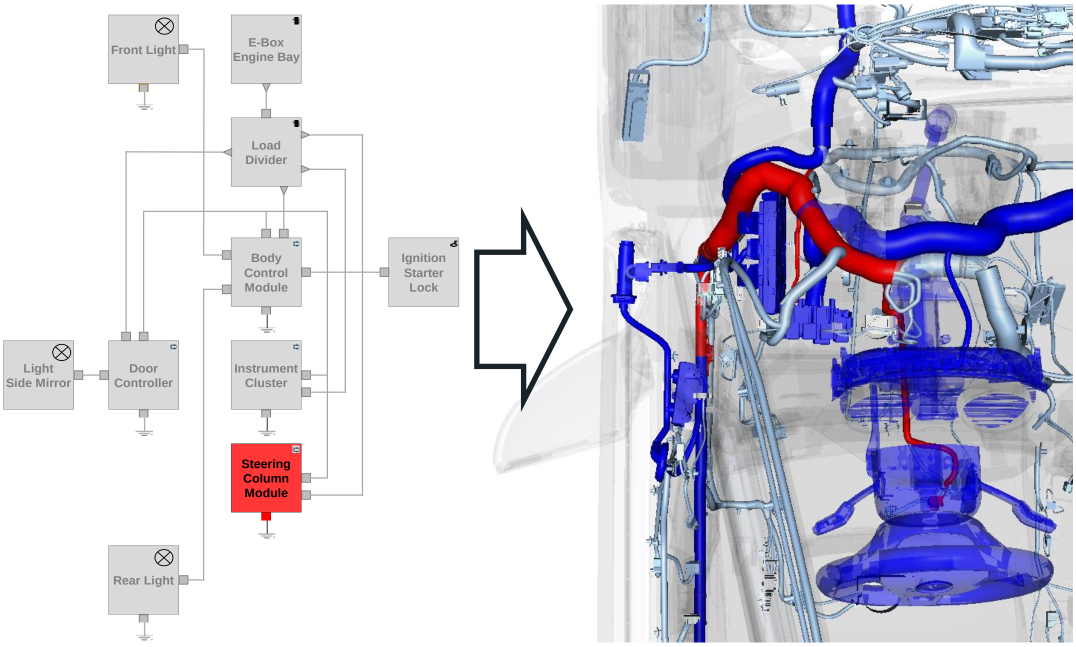

Figure 65 – The visualization (right) with the novel method provides information about the actual routing of the grounding connection which are not visible in the function architecture diagram (left)

In this example, the 3D visualization shown on Figure 65 (right) enables an efficient localization of the actual grounding connection and thus can beneficially assist function testers and service technicians in tracing affected wires and related function-oriented information.

6.2.3 Function-oriented Development and System Design

The addition of methods for a function-oriented 3D visualization to development processes adds valuable new options for development tasks. For example, it enables to directly incorporate statements and information on the spatial data such as distribution of wires and networks at the stage of architecture design, which can help to spot issues at earlier development stages and to increase quality of the architecture design and related decision-making. The following example explores benefits of the proposed 3D methodology by discussing a case that involves a typical function-related network.

A *controller area network* (CAN) is a field bus that is used for the communication between automotive systems and function components. CAN busses are widespread in distributed embedded systems due to their electrical robustness, low price, and deterministic access delay [183]. Automotive CAN busses are frequently implemented using a star topology which has some advantages in case of failure propagation regarding particular components. For instance, if there is a communication issue due to a software error in a particular component, the other components of the bus are still able to communicate properly. However, physical failure of a component may still influence all other members of a CAN bus. So, it is possible that a failure of a vehicle function can be caused by a short circuit in a system component within the bus network that is not directly related to the function at all and which is not necessarily expected at first sight. In this case, information about the location and distribution of the complete CAN network are necessary for further investigations.

This example uses the same basic architecture as it has been used in the previous section's example (Figure 65, left). In contrast, however, the architecture diagram for this use case has been extended by adding all other members of the involved CAN network (see Figure 66, left).

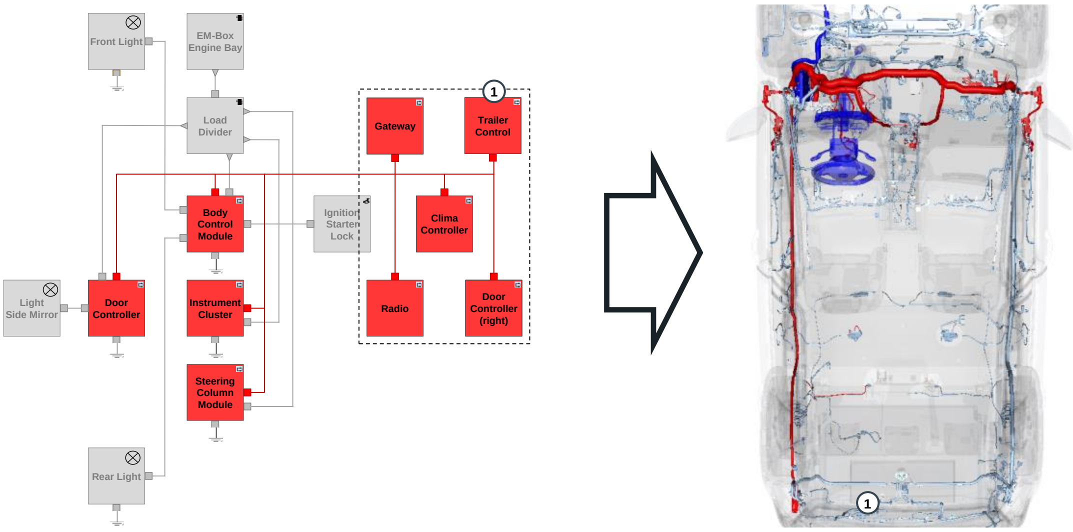

Figure 66 – A function architecture diagram (left), including all other controllers that are also connected to the involved CAN-bus network (in the dotted-line box)

In this use case, the 3D visualization highlights the full CAN network in red as shown in Figure 66 (right). While this use case can also be of benefit for service applications, for this example, a particular focus is set on applications in the development process. In that regard, a significant issue in a function-oriented development is that system designers of network architectures are usually not aware of the distribution of the designed networks in actual car configurations, at least not at early stages of development. Moreover, it needs to be recaptured that the topology of architecture diagrams usually does not represent the topology of components in an actual vehicle assembly because it is primarily designed after functional preferences and generically designed to be valid for many different vehicle derivatives. Figure 66 clearly shows that the diagram on the left does not provide any information on the actual spatial location of the wires. For example, the *Trailer Control* component is located in the upper right of the diagram space. In the physical vehicle assembly, however, as it becomes visible with the function-oriented visualization, the component is actually located in the left rear corner of the car and the wiring is routed from the frontend to the left rear as well.

Notably, as it has been pointed in Section 6.1.1, the function-oriented visualization is capable of providing a spatial view on the data from any needed perspective. This can be a significant benefit for more complex use cases. For example, Figure 67 illustrates an alternative perspective on the same set of data.

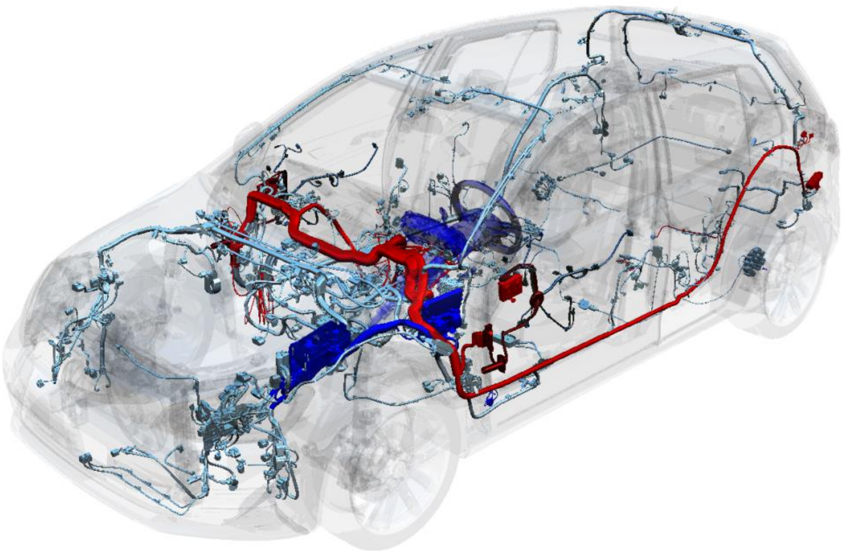

Figure 67 – Visualization based on the novel 3D method which transparently shows the complete CAN network distribution in the vehicle

The 3D methodology also enables statements on the lengths of network wires and their distribution in the vehicle. In automotive development, for physical reasons, wire lengths in networks are limited to avoid errors in synchronicity due to long runtimes and uncontrolled communication caused by reflections and parallelism of signals. Function-oriented visualization enables developing engineers to incorporate information about the summed length and spatial distribution into the architecture design so that potential errors can be spotted and avoided at early steps of development. Considering especially different variants and configurations, an approved architecture that works well in a small car does not necessarily fit in a large vehicle since wire segments can exceed the limits. Therefore, function-oriented 3D visualization helps in applying and validating existing architectures for different vehicle configurations and variants. In this regard, it can also be a valuable field of future work to implement automatic measurements of 3D data in order to provide quick statements on relevant spatial vehicle information.

Moreover, function-oriented 3D visualization enables statements on potential risks due to information access about which areas of a car may affect particular functions on crash situations (also see Chapter 6.1.5 and Chapter 7.2). For instance, sticking to the example of the left direction indicator function, Figure 66 (right) illustrates the full CAN network revealing wires routed to the left rear corner including the *Trailer Control* component. Thus, it is quickly visible that a rear-end collision (e.g. in a parking situation) can theoretically cause a short circuit in the Trailer control component and thus, for example, this collision may also cause a failure of the left *direction indicator* function. So, the proposed 3D visualization method provides many benefits for the development of function architecture networks, especially if accurate spatial localization of components is required.

6.2.4 Complementing Wiring Diagrams in Automotive Development

In automotive function-oriented development, wiring diagrams are used in the design of mechatronic systems. The exemplary use case in this section makes a limitation of such wiring diagrams apparent and, similar to the use case discussed in the previous section, it shows how the proposed function-oriented 3D methodology can assist in development processes to overcome limits of current methods such as wiring diagrams. The use case is based on the example of developing a function for a vehicle with notchback. While being a hypothetical example, it is actually based on a real-world issue that had been arisen during development of a vehicle.

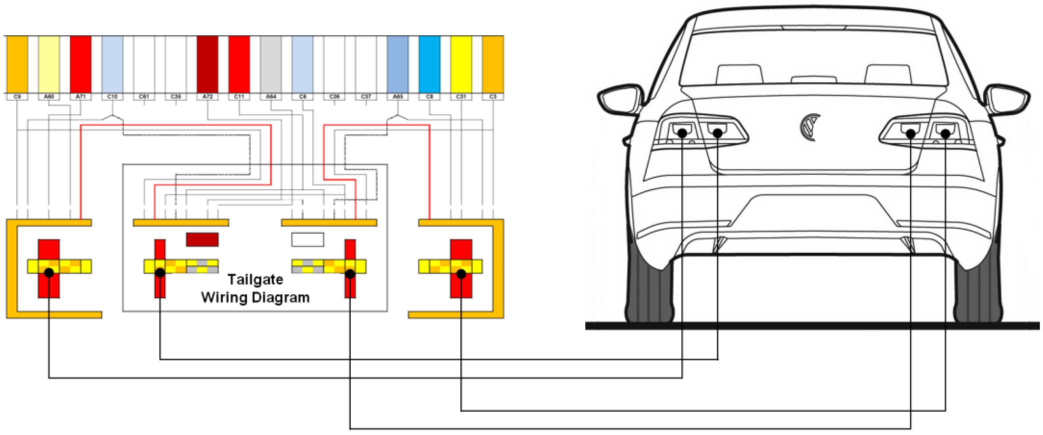

Figure 68 – Tailgate backlights in car with notchback

Figure 68 shows an exemplary but realistic wiring diagram as it could be used during development of rear lights for a vehicle with notchback. There is a pair of two back lights on each side, left and right. On each side, one of the two lights is installed in the rear gate. In the wiring diagram, the connections between the two lights of each pair are highlighted in red color. Notably, in the diagram, both connections appear to be approximately of the same length while the right connection is slightly shorter than the left one. This information, however, is misleading because the wiring diagram actually does not provide any information about the actual, physical wire lengths or their distribution in the real vehicle assembly.

The lack of information that is noticed for the wiring diagram can be filled by the function-oriented 3D visualization. So, Figure 69 illustrates a function-oriented 3D visualization based upon the integrated 3D CAD and architectural data that highlights the left wire connection in red color. This visualization complementarily provides the information on the actual spatial distribution of the wire. With this information, it is easily notable that the connection between the left-sided rear lights is comparatively short and exclusively distributed in the left back of the car.

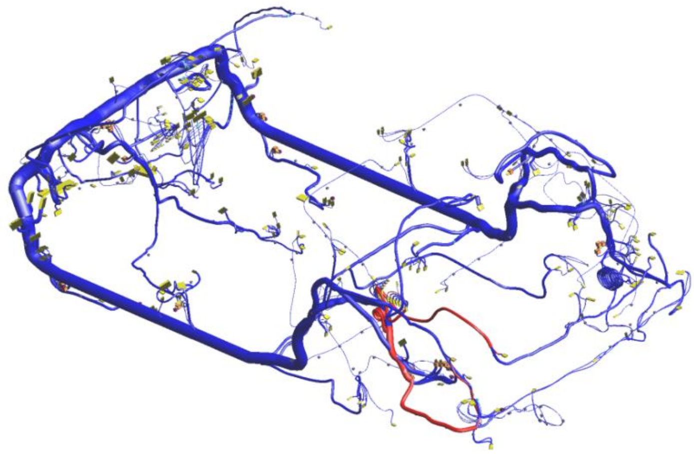

Figure 69 – A function-oriented visualization which highlights the wire connection between the left-side rear lights within a vehicles wiring network

If no further data on the spatial distribution is available, an engineer might assume that the wiring between the right-sided rear lights is implemented analogously. Interestingly, however, in the given example, the right wiring connection is actually distributed completely different as it is routed through the whole vehicle as shown in Figure 70-.

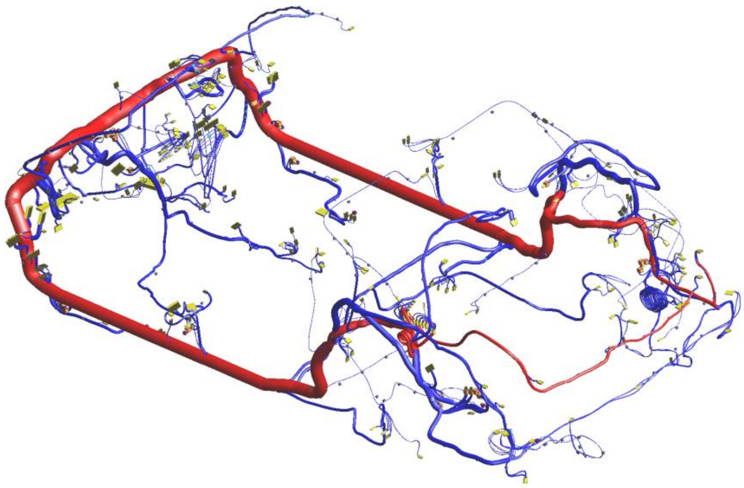

Figure 70 – A function-oriented visualization which highlights the wire connection between the right-side rear lights within a vehicles wiring network

This use case demonstrates that actual wiring distributions can be easily noticed with the spatial visualization. So, the function-oriented method enables quick identification of potential issues which would not have been detectable with the wiring diagram on its own. For instance, with this additional information, it is even easily recognizable that a line break in the car frontend can interrupt the connection between the right-sided rear lights. This issue would not have been recognizable looking at the wiring diagram. Thus, it can be argued that function-oriented 3D methods can provide a significant benefit in support of automotive development processes.

6.2.5 Integration Management of Function Controllers

In the context of function-oriented development, vehicle development processes usually involve predefined sets of milestones at which particular functionalities are to be available in the vehicle project. The term of *integration management* (IM) can be used to describe such activities which are supposed to control and coordinate actions related to such milestones and which aim to provide valid system configurations at determined dates so that, for example, validation and testing is possible in time and, as a whole, the vehicle is incrementally integrated.

The verifiability and testability of a vehicle function depends on the state of development of its particular controllers. Controllers, however, are usually frequently changing during the development process and evolving due to changes in their software implementation. Notably, a vehicle function may not be properly testable if a single component is of insufficient maturity. Therefore, integration management processes must track different aspects of such components along PLM milestones. For example, such aspects can include *state of implementation, completeness of tests* and *test maturity*.

In the following, an exemplary use case is explored which focuses on integration management of ECUs, using 3 stages of integration, respectively milestones (a real-world scenario would likely involve more than 3 milestones). In this hypothetical example, a set of ECUs is chosen which is typically involved within an automotive start-stop system. At each of the 3 milestones, the current state of the *test maturity* of each ECU is recorded as a percentage value. The hypothetical data used in this example is shown in Table 1.

Table 1 – Exemplary test maturity progress of a group of ECUs, over a period of 3 milestones

ECUs of an automotive Start-Stop function	Test Maturity (%)		
	Milestone #1	Milestone #2	Milestone #3
Airbag Module	0	0	100
Body Control Module	10	50	100
Central Locking Module	0	0	100
Instrument Cluster Module	5	60	100
Engine Module	0	30	100
Multifunction Module	0	0	100
Park-Assist Module	15	80	100
Doors Module	0	0	100

The function-oriented data integration concept proposed in this thesis allows to integrate specific data like such as illustrated in Table 1 with function architecture data (see Chapter 5.2) and 3D CAD data. Thus, it is possible to utilize this data within a 3D visualization tool so that new possibilities for data visualization are enabled. In context of this particular example, the test maturity status progress of the different ECUs can be visualized. This use case also provides an application of the function-oriented analysis methodology that has been proposed in Section 6.1.2. For this example, a color code is used which maps test maturity percentage values to colors as shown in Figure 71.

0 - 20%	21 - 40%	41 - 99%	100%

Figure 71 – Test maturity color code

With the novel 3D methodology, a visualization can be created quickly which highlights the status of the test maturity of the function ECUs listed in Table 1. Figure 72 shows a progress of the test maturity over the 3 milestones. In comparison to a traditional tabular view, the 3D visualization adds the information of the spatial distribution of these components. So, for example, a manager can quickly recognize the location of those components which have still problems with test maturity.

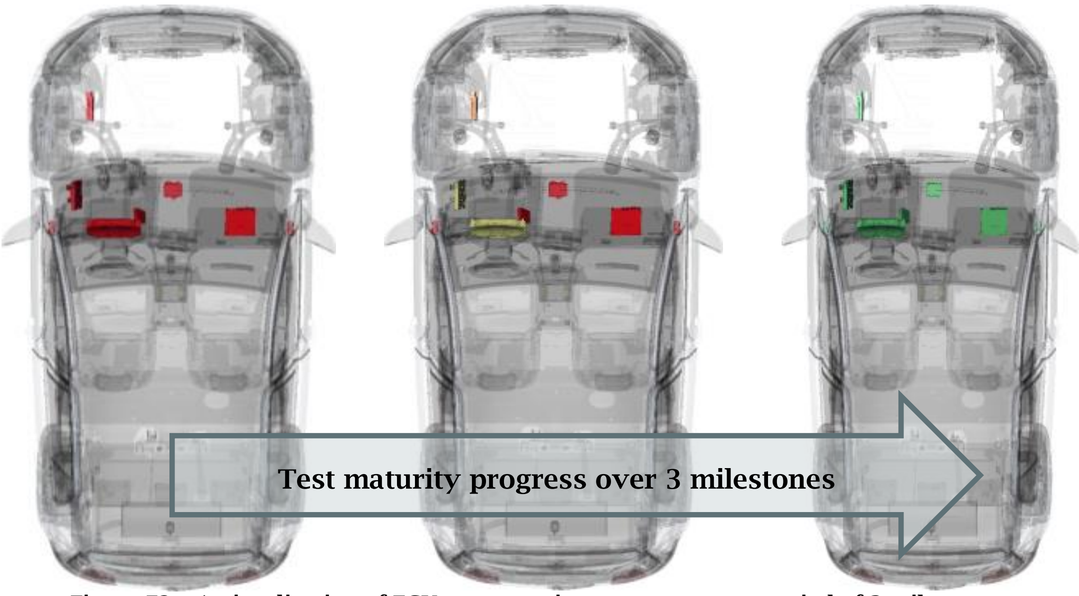

Figure 72 – A visualization of ECU test maturity progress over a period of 3 milestones

6.2.6 Visual Communication and Documentation

Proper communication is a key requirement for efficient industrial collaboration. This observation especially applies for a function-oriented development because of both the significant product complexity and the resulting necessity for interdisciplinary collaboration. Function-oriented collaboration requires that relevant information, like concepts, technical descriptions and architecture diagrams, are properly communicated across multiple different domains within the automotive PLM. For example, function architecture descriptions are not only used in development but they are also utilized in multiple other domains such as quality assurance, production and customer service.

It is well known that visual communication is by far the most dominant learning mode [184]. In addition, considering globalization, the benefits of visual communication can be even more crucial when native languages are different among multiple collaborators. Currently, in many domains of automotive development, visual communication of function-architectures is exclusively based on architecture diagrams. However, it has been shown in the previous sections that such diagrams do not provide any information on the spatial distribution and location of function-related components and connections in vehicle assemblies. Therefore, I suggest the use of function-oriented visualization to enhance communication of function architectures in order to make communication much more efficient and reduce the probability of misunderstandings.

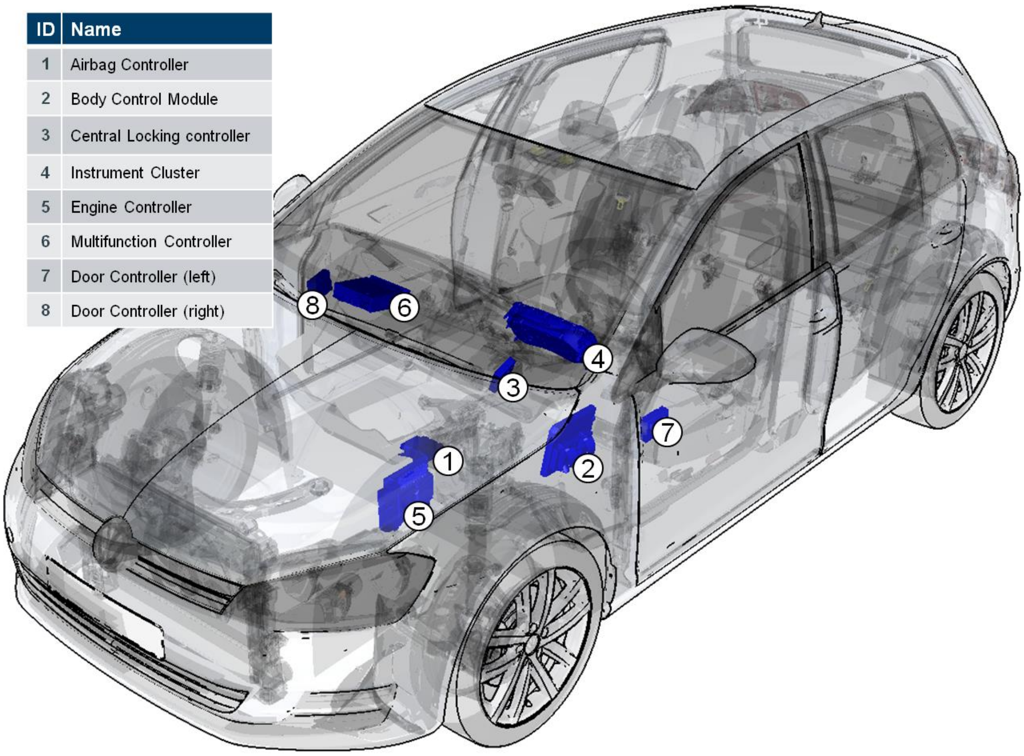

Figure 73 – The 3D visualization method can be used to create documentation pictures with the added benefits of utilizing real vehicle data and without the need for error-prone manual data preparation

I argue that function-oriented visualization enables an efficient visual communication of function architectures as well as an efficient workflow to create the required artifacts. For instance, Figure 73 illustrates an exemplary figure as it could be used for a system documentation. The figure highlights all system controllers of an automotive start-stop system, showing their names and positions within the vehicle. Especially for such tasks which require an accurate understanding of spatial positions of function elements, a 3D visualization helps to better understand and recognize technical facts related to complex function architectures. For example, such tasks are to found in automotive development, service and education. For these tasks, artifacts like documentations, pictures and videos can improve communication of related data. Currently, such artifacts need to be created with manual workflows so, for example, that respective parts in the 3D data need to be colored by hand. In contrast, the function-oriented 3D methodology has significant advantages over a manual workflow. First, the method is able to generate artifacts like pictures and videos based on real function-oriented product data. Thus, it enables an additional method for verification of particular product data which would not be possible if the visualization is created manually. Second, due to the use of real product data, the function-oriented visualization provides accurate results where a manual data preparation would be prone to errors. Third, the creation of the output work products can be easily automated and so can be performed much faster.

While static images and videos are two obvious types of media artifacts that can be generated with function-oriented visualization, a significant advantage of the proposed methodology is that it also allows for interactive, real-time 3D applications. For example, the option for real time interactions can improve learning effects in educational use cases and it can be beneficial for marketing use cases. For instance, to support exhibitions, interactive vehicle-exploration applications can be created based on real product data that is already available, without the need for further efforts to re-create data. Those and other applications provide additional examples which show how capabilities and advantages of the 3D visualization methodology can be exploited.

6.3 Summary

In this chapter, novel function-oriented 3D methods have been proposed and evaluated in different automotive use cases within automotive product lifecycle management. Figure 74 illustrates typical steps of the automotive PLM and highlights those steps in blue, which can particularly benefit from the function-oriented 3D methods, namely design, development, validation, manufacturing and support. Those steps are prone to benefit the most from the function-oriented 3D methods because related tasks and workflows require recognition and understanding of spatial distributions of function-oriented data.

Figure 74 – Typical steps within automotive product lifecycle management. Those steps which can draw most benefit from the function-oriented 3D methods are highlighted in blue.

Summarizing, primary advantages of the function-oriented 3D methods are:

- An efficient solution for precise and quick localization of components and wires related to particular vehicle functions, and related data within 3D CAD-based vehicle geometries (GFLT).
- Advanced analysis capabilities (e.g. highlight function elements matching particular conditions) and automated analyses of function architecture distributions in specific areas (like crash-relevant zones).
- A novel option for interactive and intuitive data exploration of function-oriented 3D data. For example, a user can easily navigate through 3D data and retrieve function-oriented information on selected elements.
- Accurate and fast generation of images based on real-product data which can be used for communication, documentation, learning, education, and service.
- Improved options for data management and navigation to work with function-oriented hierarchies.

7 Evaluation of the function-oriented 3D Methodologies with User Studies

This chapter presents a set of user studies which evaluate the proposed function-oriented 3D methodology in terms of time-efficiency, accuracy and usability.

7.1 Evaluation of the function-oriented 3D Method for GFLT Performance

7.1.1 Introduction

In Chapter 2.5, the GFLT has been introduced and it has been argued that it is a frequent and important task in function-oriented design. In order to further work out the capabilities of the function-oriented 3D methods to support this task, this section presents a user study which evaluates and compares the proposed novel function-oriented 3D methodology against a traditional method in particular terms of its capabilities for *time-efficient* and *accurate* GFLT performance [185]. In addition, a usability study is presented which assesses and compares the subject's subjective perception of the novel 3D method and a traditional method with respect to usability and task suitability.

7.1.2 Task-Peformance Evaluation of the 3D Method

This study assesses and compares two different methods for performing the GFLT by measuring task completion time and task solution correctness. The first method (A) is based on using conventional wiring harness diagrams and the second method (B) utilizes the function-oriented 3D visualization method which is proposed in this thesis. Synonymously, in the following, method A is also called the *traditional method* or *wiring harness method* and method B is called the *3D method*. Both methods are described in detail further down.

The study used a between-group-design and involved a total of 26 subjects, which were equally split into two groups, A and B. All subjects of group A had to use the traditional method (A), while all subjects of group B had to use the 3D method (B). The subjects were aged between 25 and 35 and had all normal or corrected-to-normal vision. All subjects have been students with technical background but without expert knowledge in any particular field of automotive development because such knowledge was not necessary in order to use the methods. Moreover, this selection of subjects ensures that the study results are not biased due to knowledge mismatches.

The primary task of the experiment involved the correct spatial identification of the electrical connections between a given set of function components for different vehicle functions. Therefore, the subjects had been given printed out function architecture diagrams which illustrate a set of function components and their connections as shown in Figure 75:

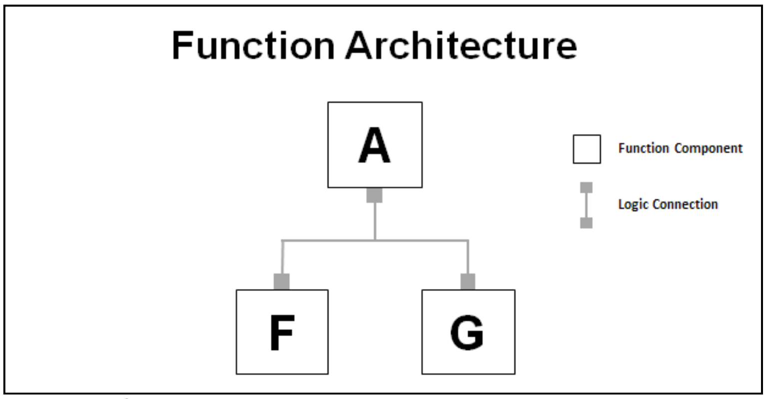

Figure 75 – A function architecture diagram as it has been used in the study

For the study, architecture diagrams of 3 hypothetical vehicle functions have been used. The first function *F1* had a relatively simple spatial distribution. It is regarded under the aspect that subjects need to go through a small learning curve to get used to the task and method. The second function *F2* is considered as the most relevant function for this research because it represents a decent task difficulty due to a non-intuitive spatial distribution of the function architecture. Also, at this point, the participants were considered to be familiar with the respective method. Thus, *F2* is defined as the *focus function* of the study. The third function *F3* involves elements of the previous functions. Thus it represents a case in which task-performance is potentially influenced by information available from the previous task. Therefore, the order of the tasks was the same for all experiments so the subjects always started with *F1*, followed by *F2* and *F3*. For the experiment task, the subjects needed to use the information from the given function architecture diagram and the respective method A or B, to identify the spatial location of the proper components and connections within an actual vehicle.

Group A had to use printed out wiring harness diagrams similar to the one shown in Figure 76. Notably, this example only shows a subsection of a full wiring harness diagram, which is about less than 20% of the full contents.

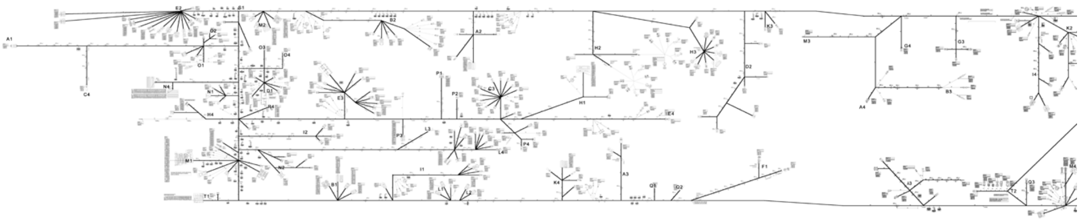

Figure 76 – A subsection of a hypothetical automotive wiring harness diagram; due to the complexity, identification of function-related components and connections is a cumbersome process using this type of data

All subjects in group A were given a brief introduction on how to use and read such wiring harness diagrams ahead of the experiments. In order to solve the study task, the subjects had to find the respective components in the wiring harness diagrams and then retrace the connections between these components by hand. Text markers could be used to highlight the identified connections as illustrated in Figure 77.

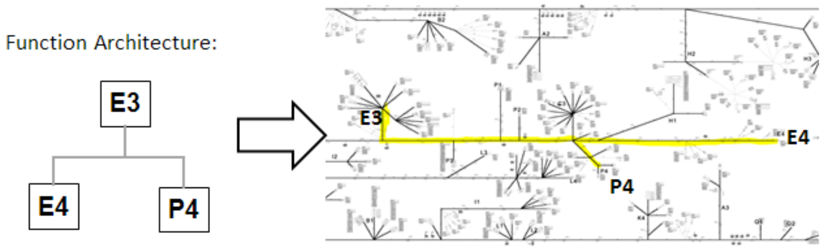

Figure 77 – Manual identification of function-related components and connections in a wiring harness diagram

In contrast, group B had to use the function-oriented 3D method which had been implemented within an established 3D visualization tool (see chapters 5 and 6). Before the study, all subjects in group B were given a short introduction on how to use the 3D tool and method. Also, a set of data was provided which included the integrated function-oriented 3D data for the respective vehicle functions used in the experiments. Figure 78 illustrates the basic user interface for the visualization tool with the loaded function-oriented 3D data.

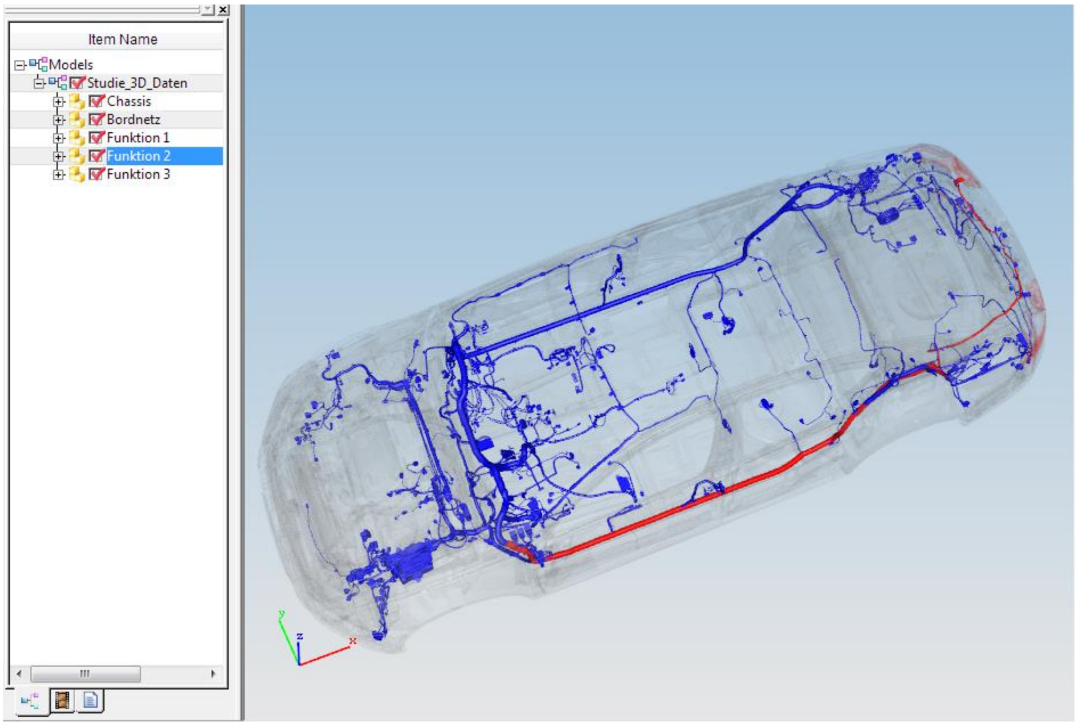

Figure 78 – The function-oriented 3D method allows to quickly highlight connections related to a particular vehicle function

Within the tool, particular vehicle functions and their elements can be quickly accessed within the hierarchical function-oriented navigation menu on the left-side of the window (see Figure 78). Selecting a particular function in the hierarchy window automatically highlights the related geometry in the 3D main window. Thus, the information about the spatial distribution of elements related to a vehicle function can be quickly obtained. Moreover, this method provides typical features of common 3D visualization tools like zooming, rotating, changing views or switching between orthographic and perspective view. Thus, a user can easily highlight and recognize the data from any perspective. Figure 79 and Figure 80 present two additional screenshots recorded with the function-oriented 3D visualization method.

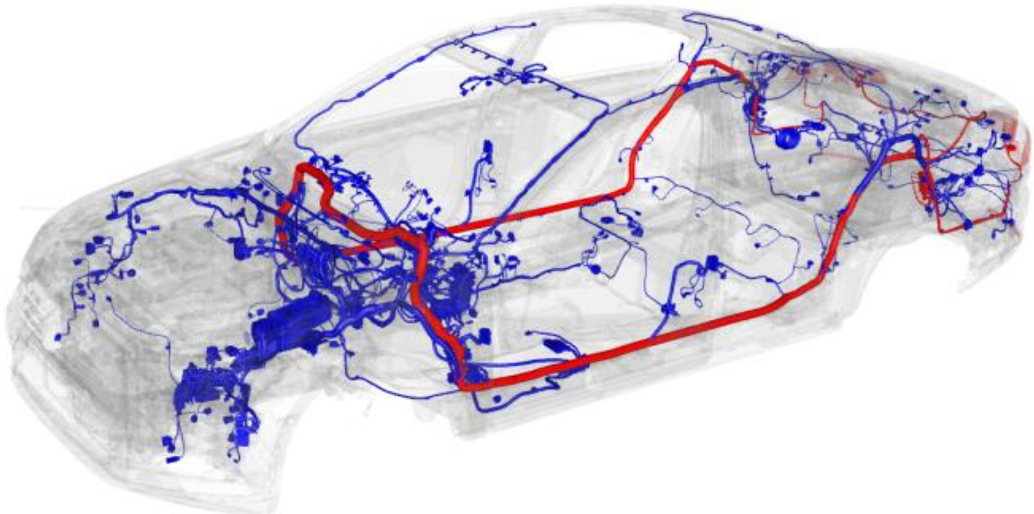

Figure 79 – Function-oriented 3D visualizations created with the 3D methodology, highlighting the relevant wires (red) of a specific function and clarifying their distribution in a particular vehicle

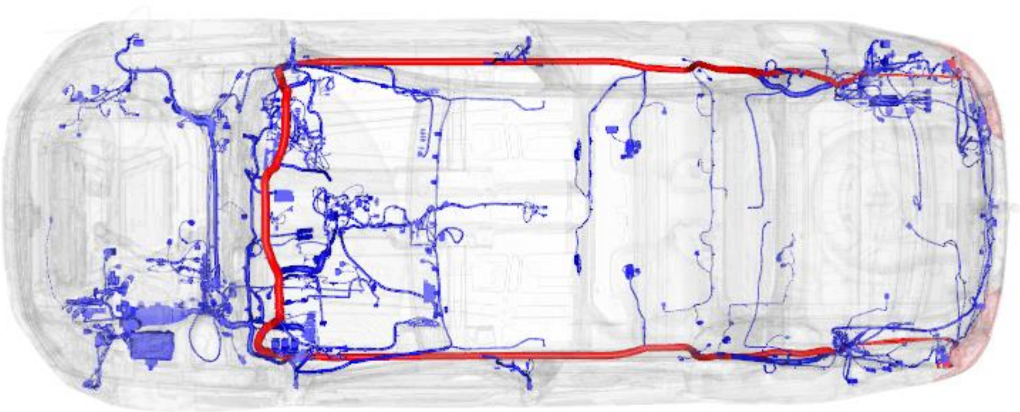

Figure 80 – Top down view of the same 3D data set as in the figure above

In order to measure solution correctness and completion time for the experiment task, for both groups, the subjects had to transfer the results of their work into a graphical template which is shown in Figure 81.

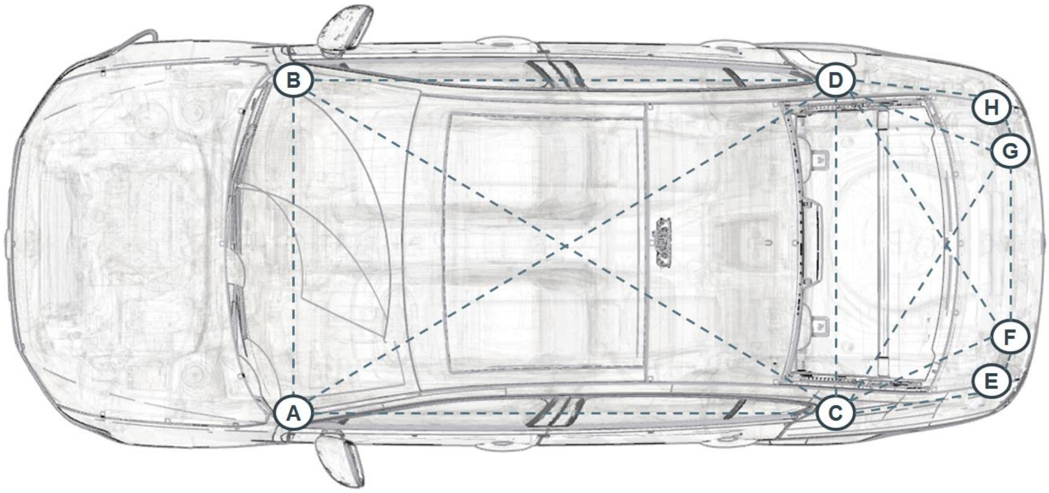

Figure 81 – The GFLT task solution template used in the study for drawing component connections

The template provides two simplification measures for the task: It shows a top-down view including X and Y dimension but not Z, and also positions of function components and valid routings are predefined (dotted lines). Leaving out the Z dimension enables comparison between the 3D method and the conventional wiring harness method because the latter is not able to provide the Z information at all. Thus, it needs to be noted that the traditional method A is only able to provide reasonable results if a top-down view is sufficient and if the spatial functional distribution involves no significant occlusions in the Z direction. This is true because, to a large extent, wiring harness diagrams involve a top-down view on the data which can be sufficient for some tasks. Otherwise, if full and precise information on spatial locations are needed for a task, it is obvious that full 3D views are necessary to recognize the spatial function architecture distribution throughout a vehicle, which will also be particularly covered in second user study in Section 7.2, including the Z dimension for accurate GFLT performance.

The top-down view is still sufficient for many use cases, and, in particular, for the tasks of this study. However, even under these simplifications, it can be demonstrated that the function-oriented 3D visualization method already provides a significant benefit for successfully performing the GFLT. Thus, it can be conjectured that it will be of even greater benefit for more complex tasks.

Figure 82 shows the complete results for the measurements of task completion time and solution correctness, including the particular results for all 26 subjects. It can be noted that there is a large difference in task completion time between F1 and F2 with method A. This indicates that the traditional method (A) requires a high learning curve. In contrast, the 3D method (B) seems to be much more intuitive as there is no significant difference between F1 and F2.

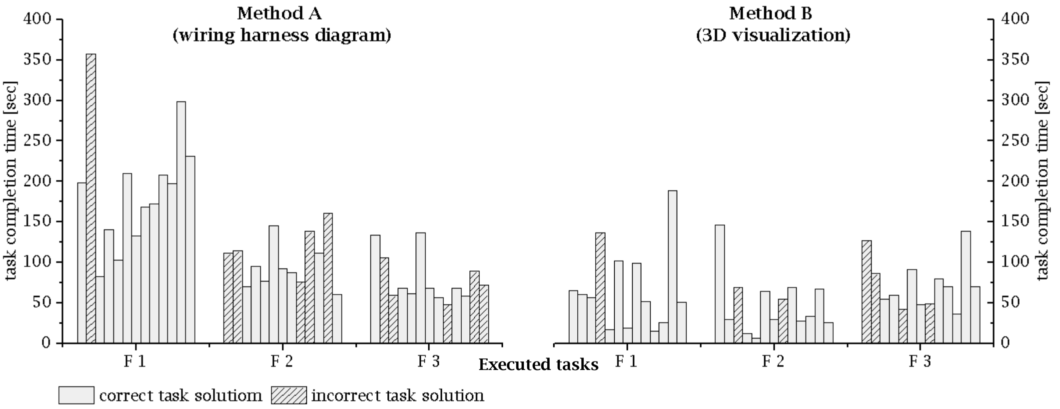

Figure 82 – Evaluation results comparing the wiring harness method to the 3D visualization method [185]. Each bar represents one subject

With the wiring harness method (A), it was found that the most time-consuming aspect of the task was the identification of the function-relevant components and their proper connections in the diagram. In addition, it was difficult for some subjects to correctly match the data for the many branches of the wiring harness diagrams. It was noticeable that those subjects who took notes and proceeded systematically (e.g. marking sections in the diagram) were significantly faster than those who did not.

With the 3D method (B), it could be noticed that some subjects spent a lot of time analyzing different 3D perspectives on the data to gain additional information that was not necessarily needed for the task solution. Thus, it can be argued that task completion time is even lower with the 3D method (B) if users are more experienced with the particular task. While some subjects had general experience with 3D tools, there were no indicators that those subjects had an advantage to solve the task with the 3D method (B). This provides evidence that it is relatively easy to use.

Figure 83 shows the comparison of both methods for all three functions, using median values in order to provide statistically robust results.

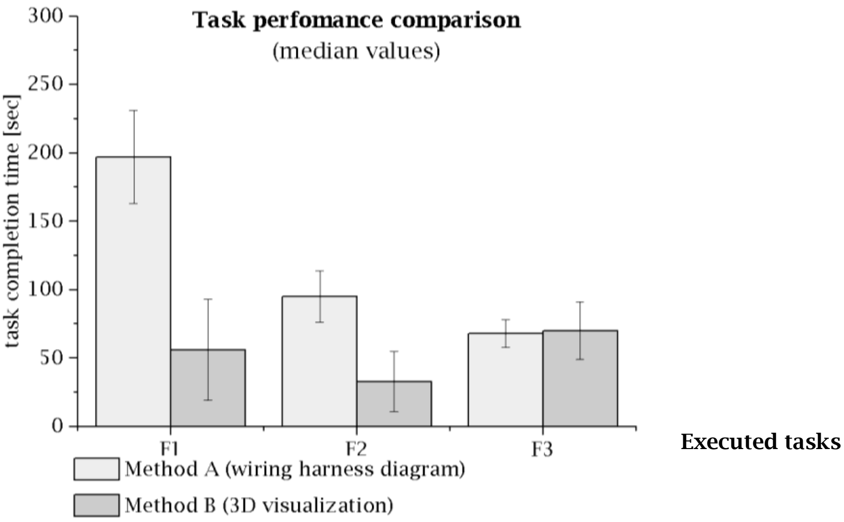

Figure 83 – Results of our study comparing task completion time (median values, incl. median absolute deviation) between the wiring harness method (A) and our 3D visualization method (B)

The results show that task completion time is significantly lower with the 3D method (B) in comparison to the wiring harness method (A) for the function F1 and, more importantly, also for the focus function F2. Compared to the wiring harness method (A), the 3D method (B) is nearly four times faster for F1 and three times faster for the focus function F2. Interestingly, the average task completion times are nearly equal between both methods for F3. Possibly, this finding can be explained by F3 being a composition of elements of F1 and F2. Thus, subjects could reuse information gained in the previous tasks to solve the task for this function, reducing the time for the search activities so that completion time is nearly equal.

Concerning task solution correctness, it was noticeable that mistakes were made with both methods (see Figure 83 and Table 2). Particularly, with the 3D method (B), a few subjects were confused with the mapping of the 3D information to the 2D template which led to some errors, especially in F3. Possibly, those mistakes can be reduced by additionally rendering metadata in the 3D visualization of the 3D method (B), which has been identified as a topic for future work.

Table 2 – Evaluation results of comparing solution correctness for both methods

Measures	***Method A*** *(wiring harness diagram)*				***Method B*** *(3D visualization)*			
	F1	**F2**	**F3**	Mean	**F1**	**F2**	**F3**	Mean
Avg. solution correctness (%)	92.3	61.5	61.5	71.8	92.3	84.6	69.2	82.1

Interestingly, the task solution correctness for F1 is equal between both methods (see Table 2) and slightly different when comparing the mean over use cases F1-F3, with method B featuring a higher solution correctness. Because F3 involves information of F2, chances are that some subjects made subsequent faults resulting from mistakes already made in F2. In any case, the most significant result is gained from the focus function F2. For this function, the results show a task solution correctness of 61.5% for the wiring harness method (A) and 83.3% for the 3D method (B). This outcome indicates that the function-oriented 3D method yields better user performance in terms of correctness.

During the study, some subjects expressed their wish to have multiple different views at the same time as well as to hide unnecessary information. These options are not possible with the harness method (A), but can be easily implemented with the 3D method (B). So, this functionality might provide a potentially beneficial feature for function-oriented 3D visualization for successful GFLT performance and it can be considered in future work.

In summary, the results of this study indicate that for GFLT performance the function-oriented 3D method is superior to the traditional wiring harness method in terms of both task completion time and solution correctness. Notably, these results apply to those cases in which a top-down view is sufficient so that wiring harness diagrams can be used. In more complex cases in which full and precise spatial information are required, wiring harness diagrams are not an alternative at all because they do not provide detailed spatial information.

7.1.3 Usability Evaluation of the 3D Method

To get a better understanding of the subject's perception of the usability and suitability of the two methods which had been compared in the previous study (see Section 7.1.2), a follow-up study has been conducted with the same 26 subjects as in the study presented before. For the evaluation method, methods such as *SUS* [186] and *AttrakDiff* [187] have been considered, but eventually, it has been decided to use a specific questionnaire which has been derived from the Nielsen usability heuristics [188,189], because the focus was not on comparison of two technical systems, but two methods with one method being computer-based and the other one being paper-based.

Each of the two groups had to fill out the questionnaire for the particular method that it had used during the previous study, which had been either the traditional wiring harness diagram method (A) or the 3D function-oriented method (B). The questionnaire included 9 particular questions using a 7-level Likert scale as follows:

1. The utilized material was helpful and relieving for performing the task
2. I cannot imagine a better method for accomplishing such tasks
3. The utilized material helped me to understand the connections
4. I was always able to keep track of my steps and I could easily memorize necessary information.
5. I did not need to repeat my steps after I made a mistake
6. I could adapt the material to my habits so that I could perform the task more efficiently
7. I easily understood how the material had to be used in order to perform the task
8. I was not overpowered by the material. The degree of detail and the presentation of information was adequate for the task.
9. I would be willing to do such tasks multiple times a day

Figure 84 illustrates the aggregated mean study results which show that both methods were graded helpful and relieving for task performance and rated equally concerning adequate degree and understandability of information and connections. However, a significant difference between the methods can be found in the question whether the subjects can imagine a better way to solve the tasks or not. While the subjects had a neutral opinion for the wiring harness method A on average, there is strong indication that most of the subjects think that the 3D method B is an ideal solution for the given task. This also is the largest deviation between the ratings of both methods.

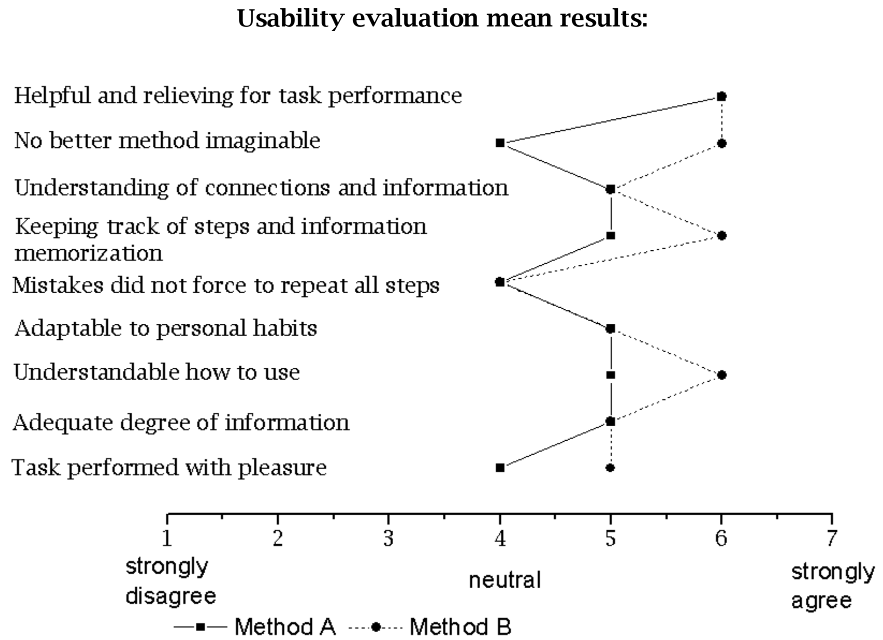

Figure 84 – The results of this evaluation show that the 3D method is superior in terms of usability compared to a traditional method using wiring harness diagrams

The results indicate that the 3D method (B) is superior to the wiring harness method (A) regarding memorization of information, which fits with prior research works on beneficial effects of spatial information for memorization tasks [190]. The 3D method (B) was also rated more understandable to use for the given task which fits with the findings from the previous study. Finally, it can be noted that the 3D method (B) has been rated more positively concerning hedonic qualities of the user experience [191].

Figure 85 shows an alternative heat map-based visualization of the study results which provides information on the number of answers in a particular direction (positive/negative) with respect to the methods heuristic attributes matching with the questionnaire. Darker squares indicate a higher number of answers in this field.

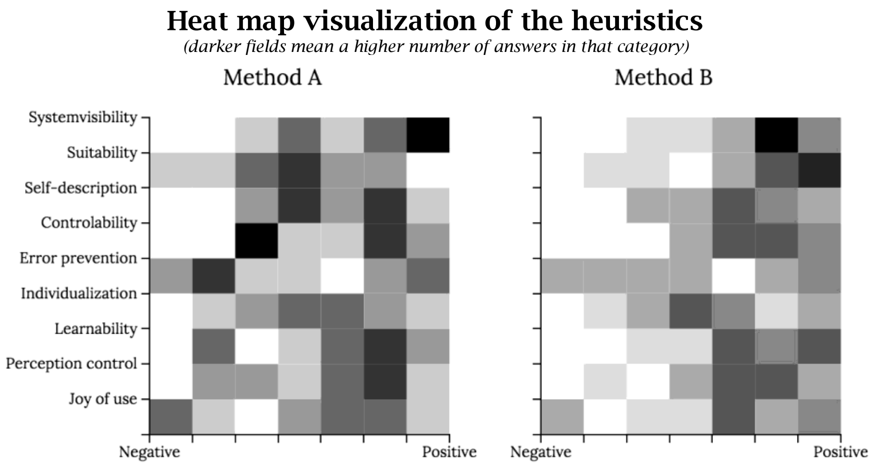

Figure 85 – Usability heuristics evaluation results – darker squares correspond with higher agreement

In summary, the subjects think that both methods provide adequate and sufficient assistance to accomplish the GFLT. However, the function-oriented 3D method is rated superior in terms of *task-suitability*, *information memorization*, *understandability*, and *joy of use*. Overall, the results provide evidence that the novel method is a suitable tool for efficient GFLT user performance and user acceptance.

7.2 Evaluating the 3D Rasterization Approach for Manual Utilization

There are a considerable number of use cases that benefit from a highest as possible accuracy in terms of GFLT performance. Especially in cases with high scene complexity and occlusions, a top-down view as it has been used in the previous study (see Chapter 7.1) might not be sufficient for proper GFLT performance. For example, an engineer might be interested in the precise overall distribution percentage of a vehicle function network throughout the vehicle in order to determine the functions overall impact or the wiring length. Or, as another example, a precise statement on the intersections of function architectures with particular vehicle zones such as crash-relevant areas might be required. Considering such use cases, the scope of the GFLT needs to be defined in a way so that it requires full spatial recognition of the function architecture distribution, including X, Y and Z dimension. Consequently, in Chapter 6.1.5, a 3D rasterization method has been proposed which addresses the above use cases. Basically, this method can be used both manually and in an automated way.

This section presents a study which aims at assessing efficiency and precision of using the 3D rasterization method in a manual way for function-oriented analyses of geometric data. Therefore, the 3D-rasterization approach has been implemented in Blender using both variants which have been discussed in Chapter 6.1.5. The first variant (variant 1) involves a grid of symmetric cells of equal sizes which flawlessly covers the complete vehicle geometry (see Figure 86, left). The second variant uses a number of specific user-defined areas in the vehicle. For this exemplary case, these areas represent crash-relevant zones in the vehicle with different deformation risks in crash situations: green (low risk), yellow (medium risk) and red (high risk) (see Figure 86, right).

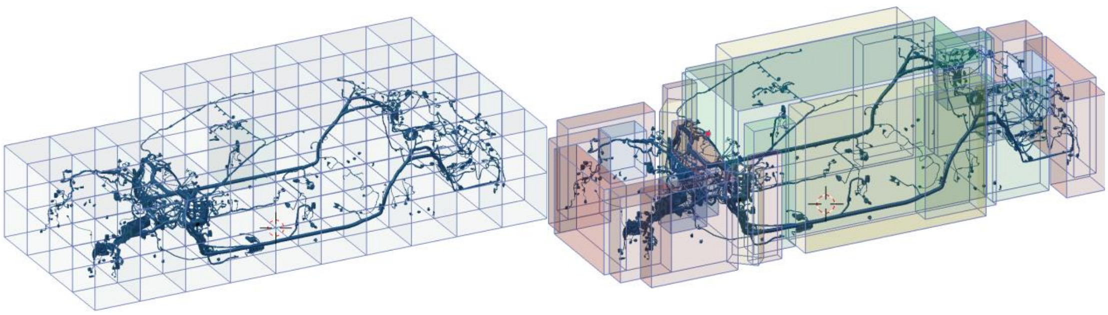

Figure 86 – Two variants of the geometric cluster approach. Variant 1 (left) uses a 3D grid of congruent cluster cells, variant 2 (right) uses volumes with different crash deformation risks. Both variants enable beneficial new features for function-oriented 3D visualization tools.

Based on these two exemplary implementations of the rasterization approach, a user study with 11 subjects has been conducted. The study goal aimed at evaluating the function-oriented 3D visualization method concerning its capabilities of providing precise statements on the spatial distribution of automotive function architectures. The subjects were aged between 20 and 40 and all had normal or corrected-to-normal vision. All subjects were given an introduction on how to use Blender in order to analyze geometric data including basic operations such as selection, navigation, zoom and rotation.

In this study, the subjects had to determine the number of cells being occupied by particular function architectures for three different vehicle functions F1, F2 and F3. Therefore, the geometric data of a full vehicle wiring harness was used in this study and for each of the 3 functions the parts occupied by this particular function were highlighted in red as shown in Figure 87.

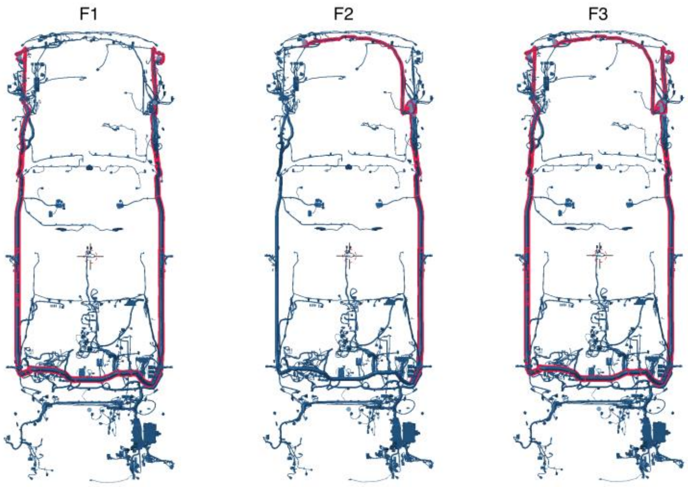

Figure 87 – Wiring harness data, highlighting elements occupied by 3 different vehicle functions in red

For each of these 3 functions, and for each of the 2 rasterization variants described above, the subjects had to perform a manual analysis of the 3D data in the 3D tool, resulting in a total of 6 tasks that each subject had to perform. The goal of each task was to count the rasterization cells which were touched by the red-highlighted function-related geometry.

For the symmetrical grid rasterization (variant 1), the subjects were asked to count the total number of cells touched by the function-related geometry. For the crash-zone areas (variant 2), the subjects were asked to count three numbers: number of red cells (1), yellow cells (2) and green cells (3) touched by the function geometry.

The study results show significant differences for task solution correctness (see Figure 88 and Table 3) between the different combinations of cluster cells and functions, especially for F1 in variant 2. This outcome indicates that the quality of the statements based on the manual 3D data analysis significantly depends on the specific locations of the cluster cells and the specific geometry, respectively on the actual scene complexity. In addition, it could be noticed that this task is considerably dependent on the user's spatial cognition abilities, colors and choices of visualization. In addition, it was noticeable that orthographic projection makes it easier to identify the occupied cells in this task in comparison to perspective projection.

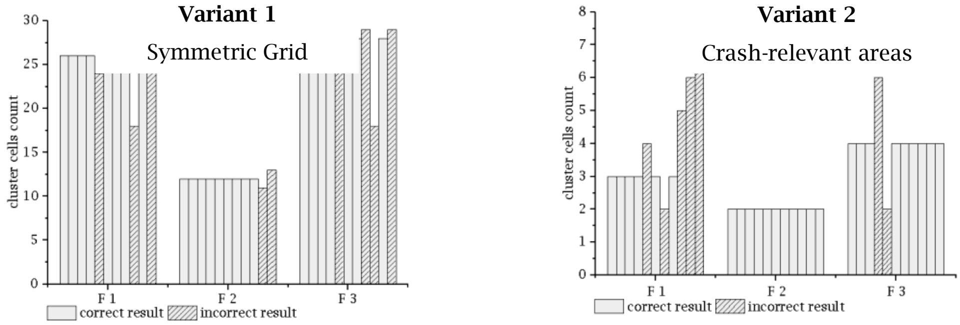

Figure 88 – The charts show the counted cells for both variants, including the information if the counts are correct or not

As it concerns the incorrect solutions, for the symmetric grid variant 1, the results show that many of the solutions only involved minor deviations from the correct solutions (deviations of +/-1 deviation in number count). Presumably, a higher geometric complexity and higher number of cluster cells increases both task completion time and error probability. Interestingly, although variant 1 involves a higher number of cells, the solution correctness is not significantly lower, which likely can be explained by a smaller scene complexity, because variant 1 involves a more structured, systematic and thus traceable design due to the symmetric cells.

Table 3 – Evaluation results of comparing task completion time and solution correctness for both methods.

Measures	***Variant 1*** *(symmetric grid)*				***Variant 2*** *(crash zones)*			
	F1	**F2**	**F3**	**Mean**	**F1**	**F2**	**F3**	**Mean**
Avg. task completion time (sec.)	108.1	45.8	63.6	*81.5*	109.3	31.8	99.3	80.1
Avg. solution correctness (%)	63.6	81.8	63.6	*69.7*	54.3	100.0	81.9	78.8

Overall, the results indicate that most of the subjects were uncertain about the correctness of their result and errors were made in all experiments with the only exception of F2 in variant 2. Thus, while it is evident that manual performance of the rasterization approach task is possible in reasonable times (see Table 3), the results show that it can be considerably prone to errors, especially when scenes are getting more complex. This conclusion is confirmed by related research findings such as [192] and [193]. Thus, it is recommendable to automate this task in order to improve accuracy and completion time and also to eliminate error probabilities. Summarizing, especially in such use cases where preferably precise statements are needed, but also in general, the function-oriented 3D rasterization approach is able to provide a new toolset for engineers to quickly perform function-oriented analyses.

8 Conclusions and Outlook

This chapter includes a summary of the thesis contributions and related conclusions and it provides an outlook to possible areas of future research.

8.1 Summary and Conclusions

Virtual reality becomes more and more popular in our modern society and it is used in many industrial and scientific sectors. Especially in the automotive industry, the development of new systems and improved workflows powered by VR and 3D CAD technologies is an increasingly important issue. Therefore, this thesis aimed at developing and evaluating novel 3D methodologies in order to answer the research question how 3D virtual reality methods can support function-oriented automotive development. In order to approach this research goal, I have proposed a concept for the consistent integration of automotive function architecture data with 3D CAD models within a representative system environment. In this context, I have defined and proposed minimum requirements, including proper data allocation, relations, completeness and granularity, in order to achieve a consistent mapping between function architecture data and 3D CAD data. In addition, I have developed an XML-based format for system-independent description and interdisciplinary cross-domain utilization and exchange of function architecture data. With the function data available in an exchangeable format, I have demonstrated that such data can be easily utilized with an established 3D visualization tool. Eventually, I have developed a prototypical implementation of the data integration concept that proves feasibility of the concept. Moreover, the applied data integration enabled me to develop and research novel function-oriented 3D methodologies and to propose, for the first time, a solution for combining function-oriented development with 3D virtual reality methods. Notably, this work fits into currently visible trends for interdisciplinary integration concepts related to virtual prototyping [7].

Based upon my prototypical implementation, I have developed and proposed a novel function-oriented 3D methodology. I have shown that this allows for new visualization techniques, manual and automated data analyses, interactive data exploration and data management. In addition, the function-oriented method can be integrated nicely into existing workflows in the automotive design process. For instance, it enables engineers to quickly and accurately recognize components and connections related to particular vehicle function architectures within virtual 3D vehicle prototypes, and to quickly and intuitively retrieve function-oriented information by exploring and selecting vehicle elements within sets of 3D CAD data. I have discussed examples on how this methodology can be added to existing and next-gen 3D visualization systems in order to improve function-oriented development processes. Moreover, I have applied the 3D methodology to a selection of automotive use cases in different areas such as development, validation and service. This research shows that the function-oriented 3D methodologies enable multiple new workflows that can be of considerable benefit in those use cases and, in general, that they are able to significantly streamline a function-oriented development and comparable approaches to systems engineering.

Finally, I have evaluated the function-oriented 3D methodology in several user studies. The results of these works show that my new methodologies provide an efficient solution to identify and recognize the spatial distribution of function architectures in specific vehicle projects. In particular, the results of the user studies indicate that the 3D method overcomes limitations of the traditional non-3D wiring harness method in terms of users' information gain and users' task performance. More precisely, I could show that the 3D method is three times more time-efficient and about 20% less prone to errors for the geometric function localization task. In addition, it closes a gap in the retrieval of function-oriented information because traditional non-3D methods such as harness diagrams do not provide 3D information at all, so that particular tasks requiring such information cannot be solved with such methods.
The usability evaluation has indicated that usability and task-suitability of the function-oriented 3D method are superior in four criteria which were rated significantly more positive than the traditional method, including *task-suitability*, *information memorization*, *understandability*, and *joy of use*. Moreover, the study has shown that my 3D methodology is more easily applicable for non-experts, which supports its interdisciplinary accessibility and usability across heterogeneous domains of automotive development.
The user study regarding tasks of detecting function architecture distribution in vehicles shows that task performance without automation of the 3D method can be prone to errors, especially in complex scenes. The automated application of the method, however, is able to provide highly accurate results in significantly fewer times. So, the highest advantage of this approach can be taken by full automation within an appropriate tool.

In conclusion, this thesis provides evidence that 3D virtual reality methods, including their particular possibilities for visualization, interaction and collaboration, are highly capable of supporting an automotive function-oriented development. Moreover, the results underline the potentials and benefits of an interdisciplinary data integration of function-oriented data and 3D CAD data. I have demonstrated that my novel methodologies can improve user performance across different use cases and domains such as development, validation and service. Thus, they can support automotive engineers in mastering challenges of current, highly-interdisciplinary function-oriented development, and, in general, the ever increasing complexity in automotive development. Notably, many other industries like shipbuilding and aircraft use comparable approaches to systems engineering which also require similar top-down breakdowns of layers such as functions, features, systems and components. Thus, most of the results of this thesis can be adapted analogously to those and other manufacturing industries. Eventually, this research work contributes to enable new and to extend existing applications for 3D virtual reality methods and it provides a ground work for further studies on function-oriented 3D methods.

8.2 Outlook and Future Work

The approach of function-oriented development is still a comparatively young concept and further work and research is necessary to fully exploit all potentials of this approach to systems engineering, in order to continuously embed the function-oriented concept throughout the full spectrum of automotive product lifecycle management. The results of this thesis suggest that there might be other automotive domains and its related data, which have not been explicitly addressed in this thesis but which can potentially benefit from similar interdisciplinary approaches as well. Thus, future work can aim to identify other domains and stakeholders which can benefit from an interdisciplinary integration of function-oriented data towards a more holistic approach to virtual prototyping. For instance, future concepts could target to include data such as simulation data to a function-oriented data integration concept. In addition, it is likely that there are even more use cases that can benefit from function-oriented 3D methodologies beyond those discussed within this thesis. For example, literature shows that 3D CAD models are already being used at university for CAx education in engineering studies [194]. Such areas of application leave interesting fields for future research.

It has been discussed that current systems, processes and tools are not able to exploit all potentials of a function-oriented data integration concept because they lack in terms of interfaces, data processing capabilities and dedicated function-oriented functionalities like those suggested in this thesis. In the future, such limitations can be overcome, for example, by the implementation of new interfaces and features to respective tools in order to further streamline function-oriented workflows. For instance, future work can focus on providing better and preferably automated interfaces between related systems. In addition, additional research should target further refinement and evaluation of appropriate function-oriented features and its addition to next-generation 3D CAD applications. Also, there is space to research new and more intuitive user interfaces to further improve usability, accessibility and overall user experience of function-oriented workflows.

Another challenging area of future research is development and research of collaborative virtual reality systems which allow multi-user exploration and utilization of function-oriented 3D virtual vehicle prototypes. This approach advances the possibilities for interdisciplinary collaboration because engineers across different domains and locations and with different areas of expertise could simultaneously view and work on a set of data. For instance, a sophisticated next-generation system could support engineers with the ability to highlight particular data and make it visible for other users so that, for example, they can mutually understand an issue and work together on a particular solution related to a vehicle function and its components. Such use cases particularly exploit the specific and sophisticated capabilities of virtual reality, such as empowering and streamlining an understanding and communication of complex data, mutual awareness, and interdisciplinary collaborations.

Appendix

A.1 Publications

Some parts of this thesis have appeared previously in the following publications:

M. Cohrs, V. Kremer, S. Klimke, G. Zachmann, *Time-efficient and accurate spatial localization of automotive function architectures with function-oriented 3D visualization*, Computer-Aided Design and Applications 13:4 (2016) 519–529.

M. Cohrs, S. Klimke, G. Zachmann, *Streamlining Function-oriented Development by Consistent Integration of Automotive Function Architectures with CAD Models*, Computer-Aided Design and Applications 11:4 (2014) 399–410.

M. Cohrs, S. Klimke, G. Zachmann, *A Methodology for Interactive Spatial Visualization of Automotive Function Architectures for Development and Maintenance*, in: G. Bebis, R. Boyle, B. Parvin, D. Koracin, B. Li, F. Porikli, V. Zordan, J. Klosowski, S. Coquillart, X. Luo, M. Chen, D. Gotz (eds.), Advances in Visual Computing, 2013, Springer-Verlag, Berlin Heidelberg, pp. 25–35.

A.2 Function-oriented XML Schema Code

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:func="VWFunctionSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="FunctionCatalog">
        <xs:annotation>
            <xs:documentation>Includes any number of functions</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Function" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Function" type="FunctionType"/>
    <xs:complexType name="FunctionType">
        <xs:sequence>
            <xs:element name="Annotations" type="textField" minOccurs="0"/>
            <xs:element name="Component" type="ComponentType" maxOccurs="unbounded"/>
            <xs:element name="Connection" type="ConnectionType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup ref="Header"/>
    </xs:complexType>
    <xs:complexType name="ComponentType">
        <xs:sequence>
            <xs:element name="Annotations" type="textField" minOccurs="0"/>
            <xs:element name="Type" type="MechatronicType"/>
            <xs:element name="MetaData" type="MetaDataType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="MappingLink" type="keyField" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup ref="Header"/>
    </xs:complexType>
    <xs:complexType name="ConnectionType">
        <xs:sequence>
            <xs:element name="Type" type="WireType"/>
            <xs:element name="Connector" type="keyField" minOccurs="2" maxOccurs="unbounded"/>
            <xs:element name="MetaData" type="MetaDataType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="MappingLink" type="keyField" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup ref="Header"/>
    </xs:complexType>
    <xs:attributeGroup name="Header">
        <xs:attribute name="id" type="keyField" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="configuration" type="keyField" use="optional"/>
    </xs:attributeGroup>
    <xs:simpleType name="MechatronicType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="sensor"/>
            <xs:enumeration value="actuator"/>
            <xs:enumeration value="controller"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="WireType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="can-bus"/>
            <xs:enumeration value="signal"/>
            <xs:enumeration value="ground"/>
            <xs:enumeration value="supply"/>
            <xs:enumeration value="other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="keyField">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="textField">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:complexType name="MetaDataType">
        <xs:attribute name="DataField" type="xs:string"/>
        <xs:attribute name="Value" type="xs:anySimpleType"/>
    </xs:complexType>
</xs:schema>
```

A.3 Example Function in XML Format (Headlamp Flasher)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<FunctionCatalog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FunctionSchemaFinal.xsd">
    <Function id="f0134" name="Headlamp Flasher">
        <Component id="s023" name="Headlamp Flasher (Button)">
            <Type>sensor</Type>
        </Component>
        <Component id="ctr012" name="Steering Pillar Module">
            <Type>controller</Type>
        </Component>
        <Component id="ctr003" name="Instrument Cluster">
            <Type>controller</Type>
        </Component>
        <Component id="ctr002" name="Body Control Module">
            <Type>controller</Type>
            <MetaData DataField="Test Maturity" Value="40%"/> <!- custom data examples -->
            <MetaData DataField="Version" Value="1.00"/>
        </Component>
        <Component id="ac005" name="Display">
            <Type>actuator</Type>
            <MetaData DataField="Size" Value="8"/>
        </Component>
        <Component id="ac006L" name="Headlight (L)" configuration="00001">
            <Type>actuator</Type>
        </Component>
        <Component id="ac006R" name="Headlight (R)" configuration="00001">
            <Type>actuator</Type>
        </Component>
        <Component id="ac007L" name="Bi-Xenon Light (L)" configuration="00002"> <!- variants -->
            <Type>actuator</Type>
        </Component>
        <Component id="ac007R" name="Bi-Xenon Light (R)" configuration="00002"> <!- variants -->
            <Type>actuator</Type>
        </Component>
        <Connection id="cn01" name="Connection1">
            <Type>other</Type>
            <Connector>s023</Connector>
            <Connector>ctr012</Connector>
        </Connection>
        <Connection id="cn02" name="Connection2">
            <Type>other</Type>
            <Connector>ctr003</Connector>
            <Connector>ac005</Connector>
        </Connection>
        <Connection id="cn03" name="Connection3">
            <Type>can-bus</Type>
            <Connector>ctr012</Connector>
            <Connector>ctr003</Connector>
            <Connector>ctr002</Connector>
        </Connection>
        <Connection id="cn04" name="Connection4" configuration="00001">
            <Type>signal</Type>
            <Connector>ctr002</Connector>
            <Connector>ac006L</Connector>
            <Connector>ac006R</Connector>
        </Connection>
        <Connection id="cn05" name="Connection5" configuration="00002">
            <Type>signal</Type>
            <Connector>ctr002</Connector>
            <Connector>ac007L</Connector>
            <Connector>ac007R</Connector>
        </Connection>
    </Function>
</FunctionCatalog>
```

A.4 PLM XML-Interface Tool

For the prototypical implementation of the data integration concept, I have developed a tool that enables a quick integration of tabular EXCEL data into existing PLM XML product structures (see Figure 89). The tool was intended to be generic and flexible to fit a broad range of use cases.

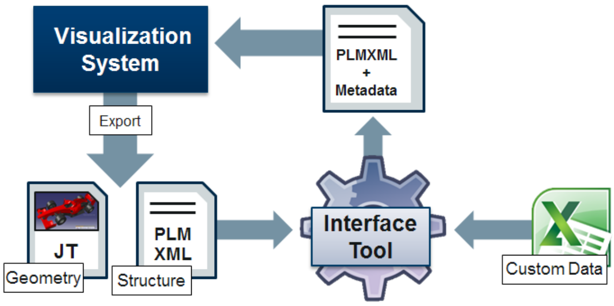

Figure 89 – Workflow for using the PLM XML interface tool to add data to PLM XML

The tool integrates tabular data into a PLM XML file using key fields for the data mapping. As a constraint, the Excel data sheet needs to be in a specific layout so that the tool is able to properly process and assign the data (see Figure 90). The first row is considered as a headline and determines the name of the key field used for the mapping as well as the names of any number of metadata fields. The first column includes identifier values which control the mapping of the data to the existing product structure.

KEY_NAME	DATA 1	…	DATA n
key_1	…	…	…
key_2	…	…	…
…	…	…	…

Figure 90 – The tabular data format required by the integration tool

The tool takes two files as input: a .PLMXML file which contains the product structure as it is present in the geometric data, and an excel sheet which contains any tabular metadata. By clicking the 'Start merge' button, the tool integrates the tabular data into the .PLMXML file. An optional check box labeled 'Exact key matching' can be set, if sub-strings are not to be allowed in the key comparison. A console window shows a report on which key fields could be successfully mapped. The tool has been implemented in Java and uses a simple graphical user interface (see Figure 91).

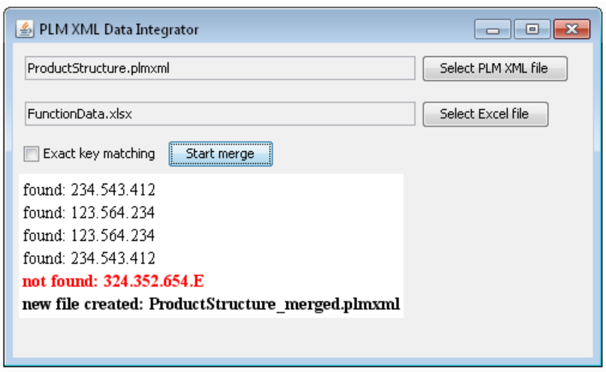

Figure 91 – Interface of the data integration interface tool

Bibliography

[1] B. Gottschalk, R. Kalmbach, *Mastering Automotive Challenges*, Kogan Page, London, 2007.

[2] SAE-China, FISITA (Eds.), Proceedings of the FISITA 2012 World Automotive Congress: Vehicle Electronics, 1st ed., Springer-Verlag, Berlin Heidelberg, 2013.

[3] B. Kaiser, B. Augustin, C. Baumann, *Von der Komponenten- zur Funktionsorientierten Entwicklung in der Funktionalen Sicherheit*, VDI-Berichte 2188 (2013).

[4] K. Revermann, Function-oriented Development, in: edaWorkshop 09 Proceedings, VDE-Verlag, Berlin, 2009.

[5] J. Weber, *Automotive development processes: Processes for successful customer oriented vehicle development*, 1st ed., Springer-Verlag, Dordrecht New York, 2009.

[6] K.L. Narayan, K.M. Rao, M. M. M Sarcar, *Computer aided design and manufacturing*, Prentice-Hall of India Privated Limited, New Delhi, 2008.

[7] M. Hirz, P. Rossbacher, J. Gulanová, *Future trends in CAD – from the perspective of automotive industry,* Computer-Aided Design and Applications 14:6 (2017) 734–741.

[8] S. Choi, K. Jung, S.D. Noh, *Virtual reality applications in manufacturing industries: Past research, present findings, and future directions*, Concurrent Engineering 23:1 (2014) 40–63.

[9] R.F. Riesenfeld, R. Haimes, E. Cohen, *Initiating a CAD renaissance: Multidisciplinary analysis driven design: Framework for a new generation of advanced computational design, engineering and manufacturing environments*, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1054–1072.

[10] S. Vajna, C. Weber, H. Bley, K. Zeman, *CAx für Ingenieure: Eine praxisbezogene Einführung*, 2nd ed., Springer-Verlag, Berlin Heidelberg, 2009.

[11] V. Duggal, A. Zoli, S. Rush, *CADD primer: A General Guide to Computer Aided Design and Drafting CAdd, CAD*, Mailmax Pub., New York, 1999.

[12] G.E. Farin, J. Hoschek, M.-S. Kim, *Handbook of computer aided geometric design*, 1st ed., Elsevier Science & Technology, Amsterdam, 2002.

[13] D.A. Madsen, D.P. Madsen, *Engineering drawing & design*, 5th ed., Delmar, Cengage Learning, Clifton Park, 2012.

[14] B. Jung, M. Latoschik, I. Wachsmuth, *Knowledge-based assembly simulation for virtual prototype modeling*, in: IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, Vol 4, 1998, Institute of Electrical and Electronics Engineers, pp. 2152–2157.

[15] G. Wang, *Definition and Review of Virtual Prototyping*, Journal of Computing and Information Science in Engineering 2:3 (2002) 232–237.

[16] F. Zorriassatine, C. Wykes, R. Parkin, N. Gindy, *A survey of virtual prototyping techniques for mechanical product development*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 217:4 (2003) 513–530.

[17] A. Gomes de Sá, G. Zachmann, *Integrating Virtual Reality for Virtual Prototyping*, in: Proceedings of DETC98 the 1998 ASME Design Engineering Technical Conferences, 1998.

[18] F. Dai, W. Felger, T. Frühauf, M. Göbel, D. Reiners, G. Zachmann, *Virtual Prototyping Examples for Automotive Industries*, in: Proceedings of Virtual Reality World'96, 1996.

[19] R. Riascos, L. Levy, J. Stjepandić, A. Fröhlich, *Digital Mock-up*, in: J. Stjepandić, N. Wognum, W. J.C. Verhagen (Eds.), Concurrent Engineering in the 21st Century: Foundation, Developments and Challenges, Springer International Publishing, Switzerland, 2015, pp. 355–388.

[20] I.J. Freeman, J.L. Salmon, J.Q. Coburn, *CAD Integration in Virtual Reality Design Reviews for Improved Engineering Model Interaction*, in: ASME 2016 International Mechanical Engineering Congress and Exposition, Vol 11: Systems, Design, and Complexity, ASME, 2016, pp. V011T15A006.

[21] O. Enge-Rosenblatt, P. Schneider, C. Clauß, A. Schneider, *Functional Digital Mock-up - Coupling of Advanced Visualization and Functional Simulation for Mechatronic System Design*, in: W. Commerell (Ed.), ASIM-Treffen 2010 der Fachgruppen: "Simulation technischer Systeme" und "Grundlagen und Methoden in Modellbildung und Simulation", 2010, Hochschule Ulm, pp. 41–48.

[22] F. Krause, H. Franke, J. Gausemeier, *Innovationspotenziale in der Produktentwicklung*, Carl Hanser Verlag GmbH & Co. KG, München, 2006.

[23] C. Schäfer, L. Voigt, T. Bruder, A. Stork, C. Clauß, A. Schneider, T. Farkas, A. Hinnerichs, *FDMU-Plattform zur Unterstützung der Produktentwicklung mechatronischer Systeme*, in: SIMVEC - Numerical Analysis and Simulation and Vehicle Engineering, 2008, Vol 2031 von VDI Berichten, 2008, VDI-Verlag.

[24] U. Seiffert, G.P. Rainer, *Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz: Prozesse, Komponenten, Beispiele aus der Praxis*, 1st ed., Vieweg + Teubner Verlag / GWV Fachverlage, Wiesbaden, 2008.

[25] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, H. Elmqvist, A. Junghanns, J. Mauß, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, S. Wolf, *The Functional Mockup Interface for Tool Independent Exchange of Simulation Models*, in: C. Clauß (Ed.), Proceedings of the 8th International MODELICA Conference, 2011, Linköping University Electronic Press, pp. 105–114.

[26] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, A. Viel, *Functional mockup interface 2.0: The standard for tool independent exchange of simulation models*, in: M. Otter and D. Zimmer (Eds.) Proceedings of the 9th International MODELICA Conference, 2012, Linköping University Electronic Press, pp. 173–184.

[27] O. Enge-Rosenblatt, C. Clauß, A. Schneider, Schneider. P, *Functional digital mock-up and the functional mock-up interface - two complementary approaches for a comprehensive investigation of heterogeneous systems*, in: C. Clauß (Ed.), Proceedings of the 8th International MODELICA Conference, 2011, Linköping University Electronic Press, pp. 748–755.

[28] W.R. Sherman, A.B. Craig, *Understanding Virtual Reality: Interface, Application, and Design*, 1st ed., Morgan Kaufmann Publisher, San Francisco, 2003.

[29] G. Lawson, D. Salanitri, B. Waterfield, *Future directions for the development of virtual reality within an automotive manufacturer*, Applied Ergonomics 53, Part B (2016) 323–330.

[30] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, *Recent advances in augmented reality*, in: IEEE Computer Graphics and Applications 21:6, 2001, Institute of Electrical and Electronics Engineers, pp. 34–47.

[31] M. Tönnis, *Augmented Reality - Einblicke in die Erweiterte Realität*, 1st ed., Springer-Verlag, Berlin Heidelberg, 2010.

[32] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, M. Ivkovic, *Augmented reality technologies, systems and applications, Multimedia Tools and Applications* 51:1 (2011) 341–377.

[33] F. Zhou, H.B.-L. Duh, M. Billinghurst, *Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR*, in: M.A. Livingston, O. Bimber, H. Saito (Eds.), 7th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2008, IEEE Computer Society, pp. 193–202.

[34] Wikipedia, Pokémon Go, available at https://en.wikipedia.org/wiki/Pok%C3%A9mon_Go (accessed on August 20, 2016).

[35] S. Nölle, G. Klinker, *Augmented Reality as a Comparison Tool in Automotive Industry*, in: C. Rutledge, R. Schoenfeld (Eds.), Proceeding of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006, IEEE Computer Society, pp. 249–250.

[36] Volkswagen AG, Virtual Technologies: MARTA – innovative service support tool for the Volkswagen XL1, available at http://www.volkswagenag.com/en/group/research/virtual-technologies.html (accessed on June 11, 2017).

[37] A. Stork, C.-A. Thole, S. Klimenko, I. Nikitin, L. Nikitina, Y. Astakhov, *Simulated Reality in Automotive Design*, in: F.-E. Wolter, A. Sourin (Eds.), Proceedings of International Conference on Cyberworlds, 2007, IEEE Computer Society, pp. 23–27.

[38] F. Purschke, M. Schulze, P. Zimmermann, *Virtual reality-new methods for improving and accelerating the development process in vehicle styling and design*, in: K. Kelly (Ed.), Proceedings of Computer Graphics International, 1998, IEEE Computer Society, pp. 789–797.

[39] N.R. Rao, *Innovation through Virtual Technologies,* in: E.V. Prasad, K. Hariharanath, R. Gopalaswamy (Eds.), Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches, 1st ed., Business Science Reference, Hershey New York, 2011, pp. 1–13.

[40] A. Schilling, S. Kim, D. Weissmann, Z. Tang, S. Choi, *CAD-VR geometry and meta data synchronization for design review applications*, Journal of Zhejiang University Science A: Applied Physics and Engineering 7:9 (2006) 1482–1491.

[41] G. Zachmann, *VR-Techniques for Industrial Applications*, in: F. Dai (Ed.), Virtual Reality for Industrial Applications, Springer-Verlag, Berlin Heidelberg, 1998, pp. 13–38.

[42] J. Bowers, S.D. Benford, S*tudies in Computer-Supported Cooperative Work: Theory, Practice and Design Human Factors in Information Technology*, Elsevier Science Ltd, Amsterdam, 1991.

[43] I. Greif, *Computer-supported cooperative work: A book of readings*, Morgan Kaufmann Publisher, San Francisco, 1988.

[44] C.A. Ellis, S.J. Gibbs, G.L. Rein, *Groupware - Some Issues and Experiences*, in: P.J. Denning (Ed.) Communications of the ACM, 34:1 (1991) 35–58.

[45] L.J. Bannon, K. Schmidt, *CSCW: Four characters in search of a context*, in: J. Bowers, S. Benford (Eds.), Proceedings of the First European Conference on Computer Supported Cooperative Work, 1989, Computer Science Company, pp1–20.

[46] S. Harrison, *Media space 20+ years of mediated life*, Springer-Verlag, London, 2009.

[47] G. Schwabe, N. Streitz, R. Unland, *CSCW-Kompendium: Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten*, Springer-Verlag, Berlin Heidelberg, 2001.

[48] Siemens PLM Software, Teamcenter, available at http://www.siemens.com/Teamcenter (accessed on February 8, 2013).

[49] ISO 14306:2017: Industrial automation systems and integration -- JT file format specification for 3D visualization.

[50] Siemens PLM Software, JT Open, available at https://www.plm.automation.siemens.com/de_de/products/open/jtopen/ (accessed on July 14, 2016).

[51] Stratasys Software, GrabCAD: Hub for digital manufacturing, available at https://grabcad.com/ (accessed on July 19, 2016).

[52] World Wide Web Consortium (W3C), W3C, available at http://www.w3.org/ (accessed on March 15, 2013).

[53] L. Wang, A.Y.C. Nee, *Collaborative Design and Planning for Digital Manufacturing*, 1st ed., Springer-Verlag, London, 2009.

[54] E.R. Harold, W.S. Means, *XML in a nutshell: A desktop quick reference*, 3rd ed., O'Reilly Media, Sebastopol, 2005.

[55] B. Gottschalk, *The automotive industry sets the course for the global economy*, in: B. Gottschalk, R. Kalmbach (Eds.), Mastering Automotive Challenges, Kogan Page, London, 2007, pp. 3–24.

[56] R. Kalmbach, *The automotive power play moves into its next round*, in: B. Gottschalk, R. Kalmbach (Eds.), Mastering Automotive Challenges, Kogan Page, London, 2007, pp. 25–45.

[57] P. Bauer, *How electronics is changing the automotive industry: from component suppliers to system partners*, in: B. Gottschalk, R. Kalmbach (Eds.), Mastering Automotive Challenges, Kogan Page, London, 2007, pp. 270–289.

[58] G. Leen, D. Heffernan, *Expanding automotive electronic systems*, Computer 35:1 (2002) 88–93.

[59] D. Politze, J. Bathelt (Eds.), *Exploitation Method for Functional Product Requirements: An Integrated Function Oriented Approach*, in: A. García-Cerezo, K Ohnishi (Eds.), Proceedings of International Conference on Mechatronics, 2009, Institute of Electrical and Electronics Engineers.

[60] K. Reif, *Automobilelektronik: Eine Einführung für Ingenieure*, 5th ed., Vieweg+Teubner Verlag, Wiesbaden, 2014.

[61] VDI-Gesellschaft, *Entwicklungs-Methodik für mechatronische Systeme*, VDI 2206, Beuth Verlag, Berlin, 2004.

[62] O. Hancu, V. Maties, R. Balan, Stan, Sergiu, Lapusan, C., *Mechatronic approach for design and control of a hydraulic 3-DOF parallel robot*, in: B. Katalanic (Ed.), DAAAM International Scientific Book 2007, 2007, DAAAM International, pp. 525–536.

[63] E.A. Lee, S.A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*, 2nd ed., MIT Press, 2017.

[64] S.C. Suh, U.J. Tanik, J.N. Carbone, A. Eroglu, *Applied Cyber-Physical Systems*, 1st ed., Springer-Verlag, New York, 2014.

[65] E.A. Lee (Ed.), *Cyber Physical Systems: Design Challenges*, in: L. O'Conner (Ed.) 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, 2008, IEEE Computer Society, pp. 363–369 .

[66] J.P. MacDuffie, Sethuraman K., M.L. Fisher, *Product Variety and Manufacturing Performance: Evidence from the International Automotive Assembly Plant Study*, Management Science 42:3 (1996) 350–369.

[67] K.S. Wever, L. Turner, *The Comparative political economy of industrial relations, 1st ed., Industrial Relations Research Association*, Madison, WI: Industrial Realtions Research Asscociation, 1995.

[68] S. Schindler, Roland Berger Consultants, *The technology challenge: progress or pitfall?,* in: B. Gottschalk, R. Kalmbach (Eds.), Mastering Automotive Challenges, Kogan Page, London, 2007, pp. 103–145.

[69] C. Ebert, C. Jones, *Embedded Software: Facts, Figures, and Future*, Computer 42:4 (2009) 42–52.

[70] R.N. Charette, *This car runs on code*, IEEE spectrum 46 (2009).

[71] G. Reinhart, O. Anton, B. Lercher, *Funktionsorientiertes Sichtenmodell für die Entwicklung mechatronischer Systeme*, in: VDI-Z Integrierte Produktion, 11-12, (2001) 67-70.

[72] V. Tanneberger, *Funktionsorientierte Entwicklung: Der Schlüssel zu Kosteneffizienz und Qualität elektronischer Systeme*, in: 13. Euroforum Jahrestagung: Elektroniksysteme in Automobil, 2009.

[73] IABG, Das V-Modell ®, available at http://www.v-modell.iabg.de/ (accessed on December 10, 2017).

[74] Volkswagen AG, Kurvenfahrlicht, available at http://www.volkswagen.de/de/technologie/technik-lexikon/kurvenfahrlicht.html (accessed on May 27, 2017).

[75] Volkswagen AG, Parking Steering Assistance, available at http://www.volkswagen.de/de/technologie/technik-lexikon/parklenkassistent_park_assist.html (accessed on May 27, 2017).

[76] H.H. Goldstine, A. Goldstine, *The electronic numerical integrator and computer (ENIAC)*, in: IEEE Annals of the History of Computing, Vol 18:1 (1996), IEEE Computer Society, pp. 10–16.

[77] S. McCartney, *ENIAC: The triumphs and tragedies of the world’s first computer*, 1st ed., Walker Books, London, 1999.

[78] M. Heilig, *Sensorama Simulator*, (Clakson, D.M.), Patent No. 3050870A, 1962 August, 28.

[79] I.E. Sutherland, *The Ultimate Display*, in: W.A. Kalenich (Ed.), Proceedings of the 1965 IFIP Congress International Federation for Information Processing, 1965, Macmillan and Co, pp. 506-508.

[80] Webster's Online Dictionary, available at http://www.websters-online-dictionary.org/ (accessed on February 22, 2012).

[81] Brockhaus Enzyklopädie: In vierundzwanzig Bänden, F.A. Brockhaus, Mannheim, 1994.

[82] D. Holmes, *Communication Theory - Media, Technology and Society*, 1st ed., SAGE Publications Ltd, London, 2005.

[83] K.M. Stanney, K.S. Hale, *Handbook of Virtual Environments - Design, Implementation, and Applications*, 2nd ed., CRC Press, Mahwah, New Jersey, 2014.

[84] Merriam-Webster Dictionary, Full Definition of virtual reality, available at http://www.merriam-webster.com/dictionary/virtual%20reality (accessed on June 11, 2017).

[85] C. Dodsworth, *Digital illusion: Entertaining the future with high technology*, 1st ed., ACM Press/ Addison-Wesley Publishing, New York, 1998.

[86] M. Slater, V. Linakis, M. Usoh, R. Kooper, G. Street, *Immersion, Presence, and Performance in Virtual Environments: An Experiment with Tri-Dimensional Chess*, in: ACM Virtual Reality Software and Technology, 1996, Association of Computing Machinery, pp. 163–172.

[87] D.A. Bowman, R.P. McMahan, *Virtual Reality: How Much Immersion Is enough?,* Computer 40 (2007) 36–43.

[88] B. Laha, K. Sensharma, J.D. Schiffbauer, D.A. Bowman, *Effects of immersion on visual analysis of volume data*, IEEE transactions on visualization and computer graphics, 18:4 (2012) 597–606.

[89] I. Cho, W. Dou, Z. Wartell, W. Ribarsky, X. Wang (Eds.), *Evaluating depth perception of volumetric data in semi-immersive VR*, in: S. Coquillart, S. Feiner, K. Kiyokawa (Eds.), 2012 IEEE Virtual Reality Workshops, 2012, IEEE Computer Society, pp. 95–96 .

[90] P. Schuchardt, D.A. Bowman, *The benefits of immersion for spatial understanding of complex underground cave systems*, in: S.N. Spencer (Ed.), Proceedings of the 2007 ACM symposium on Virtual reality software and technology, 2007, Association of Computing Machinery, pp. 121–124.

[91] C. Regan, *An investigation into nausea and other side-effects of head-coupled immersive virtual reality*, Virtual Reality 1:1 (1995) 17–31.

[92] J.J. Gibson, *The ecological approach to visual perception*, revised ed., Routledge, London, 1986.

[93] B.G. Witmer, M.J. Singer, *Measuring Presence in Virtual Environments: A Presence Questionnaire*, Presence Teleoperators and virtual Envoirments 7:3 (1998) 225–240.

[94] M. Slater, *Measuring Presence: A Response to the Witmer and Singer Presence Questionnaire*, Presence: Teleoperators and Virtual Environments 8:5 (1999) 560–565.

[95] M.V. Sanchez-Vives, B. Spanlang, A. Frisoli, M. Bergamasco, M. Slater, *Virtual hand illusion induced by visuomotor correlations*, PloS one, 5:4 (2010) e10381.

[96] M. Slater, A. Steed, *A Virtual Presence Counter*, Presence: Teleoperators and Virtual Environments 9:5 (2000) 413–434.

[97] A. Maselli, M. Slater, *The building blocks of the full body ownership illusion*, Frontiers in human neuroscience 7:83 (2013) 83.

[98] D. Friedman, R. Pizarro, K. Or-Berkers, S. Neyret, X. Pan, M. Slater, *A method for generating an illusion of backwards time travel using immersive virtual reality—an exploratory study*, Frontiers in Psychologie 5:943 (2014) 183.

[99] J. Steuer, *Virtual Reality: Dimensions Determining Telepresence*, Journal of Communication 42:4 (1992) 73–93.

[100] D. Talabă, A. Amditis, *Product engineering: Tools and methods based on Virtual Reality*, 1.ed., Springer Science + Buisness Media B.V., Berlin Heidelberg, 2008.

[101] J. Han, L. Shao, D. Xu, J. Shotton, *Enhanced computer vision with microsoft kinect sensor: A review*, IEEE transactions on cybernetics 43:5 (2013) 1318–1334.

[102] Z. Zhang, *Microsoft kinect sensor and its effect*, IEEE multimedia 19:2 (2012) 4–10.

[103] P.R. Desai, P.N. Desai, K.D. Ajmera, K. Mehta, *A review paper on oculus rift-a virtual reality headset*, International Journal of Engeering Trends and Technology 13:4 (2014) 175–179.

[104] W. Felger, T. Fröhlich, M. Göbel (Eds.), *Techniken zur Navigation durch Virtuelle Welten*, Virtual Reality (1993) 209–222.

[105] G. Zachmann, *Virtual Reality in Assembly Simulation -- Collision Detection, Simulation Algorithms, and Interaction Techniques*, phD Thesis, 2000.

[106] K. Pimentel, K. Teixeira, *Virtual reality: Through the new looking glass*, 1st ed., TAB Books Inc, New York, 1993.

[107] H. Chen, H. Sun, *Body-Based Haptic Interaction Model for Touch-Enabled Virtual Environments*, Presence: Teleoperators and Virtual Environments 15:2 (2006) 186–203.

[108] G.C. Burdea, *Haptics issues in virtual environments*, in: A. Rawlinson (Ed.), Proceedings Computer Graphics International 2000, IEEE Computer Society, pp. 295–302.

[109] H. Sun, H. Chen, *Soft-Touch Haptics Modeling of Dynamic Surfaces,* in: N.R. Rao (Ed.), Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches, 1st ed., Business Science Reference, Hershey New York, 2011, pp. 14–35.

[110] M.S. Pinho, C. M. D. Freitas, *Cooperative and Simultaneous Object Manipulation in Collaborative Virtual Environments*, Journal of the Brazilian Computer Society 14:2 (2008) 53–67.

[111] G.C. Burdea, P. Coiffet, *Virtual reality technology*, 2nd ed., Wiley-IEEE Press, New Jersey, 2003.

[112] R. Dörner, W. Broll, P. Grimm, B. Jung, *Virtual Reality und Augmented Reality (VR/AR) - Grundlagen und Methoden der Virtuellen und Augmentierten Realität*, 1st ed., Springer-Verlag, Berlin Heidelberg, 2016, pp. 30–37.

[113] N.N. Latypov, *Method for tracking and displaying user's spatial position and orientation*, (N. Latypov), Patent No. US6005548 A, 1999 august, 14.

[114] A.M. Wichansky, *User benefits of visualization with 3-D stereoscopic displays*, in: J.O. Merritt, S.S. Fisher (Eds.) Proceedings Stereosopic Displays and Applications II, Vol 1457, 1991, Society of Photo-Optical Instrumentation Engineers.

[115] Z.X. Jin, Y.J. Zhang, X. Wang, T. Plocher (Eds.), *Evaluating the Usability of an Auto-stereoscopic Display*, in: J.A. Jacko (Ed.) Human-Computer Interaction. Interaction Platforms and Techniques. HCI 2007. Lecture Notes in Computer Science, Vol 4551, Springer, Berlin Heidelberg, pp. 605–614.

[116] M. Tsuboi, S. Kimura, T. Horikoshi, *An objective and subjective evaluation of an autostereoscopic 3d display*, in: Proceeding of CHI '09 Extended Abstracts on Human Factors in Computing Systems, 2009, Association of Computing Machinery, pp. 3577–3582.

[117] N.A. Dodgson, *Autostereoscopic 3D Displays*, IEEE Computer 38:8 (2005) 31–36.

[118] N.A. Dodgson, J.R. Moore, S.R. Lang, *Multi-View Autostereoscopic 3D Display*, in: IBC '99 International Broadcasting Convention, 1999, pp. 497–502.

[119] H. Stolle, J. Olaya, S. Buschbeck, H. Sahm, A. Schwerdtner, *Technical solutions for a full-resolution auto-stereoscopic 2D/3D display technology*, in: Proceedings of Stereoscopic Displays and Applications XIX, Vol 6803, 2008, Society of Photo-Optical Instrumentation Engineers, pp. 68030Q-68030Q-12.

[120] E.M. Kolasinski, *Simulator Sickness in Virtual Environments*, U.S. Army Research Institute for the Behavioral and Social Sciences, 1995.

[121] A. Kulik, A. Kunert, S. Beck, R. Reichel, R. Blach, A. Zink, B. Froehlich, *C1x6: a stereoscopic six-user display for co-located collaboration in shared virtual environments*, ACM Transactions on Graphics, 30:6 (2011) 188:1–188:11.

[122] P. Santos, A. Stork, T. Gierlinger, A. Pagani, B. Araújo, R. Jota, L. Bruno, J. Jorge, J. Madeiras Pereira, M. Witzel, G. Conti, R. de Amicis, I. Barandarian, C. Paloc, O. Machui, J.M. Jiménez, G. Bodammer, D. McIntyre, *IMPROVE: Collaborative Design Review in Mobile Mixed Reality*, in: R. Shumaker (Ed.), Virtual Reality: Second International Conference, ICVR 2007, Springer-Verlag, Berlin Heidelberg, 2007, pp. 543–553.

[123] G. Moreau, P. Fuchs, A.J. Woods, J.O. Merritt, S.A. Benton, M.T. Bolas, *Stereoscopic displays for virtual reality in the car manufacturing industry: application to design review and ergonomic studies*, in: Proceedings of Stereoscopic Displays and Virtual Reality Systems IX, Vol 4660, 2002, Society of Photo-Optical Instrumentation Engineers, pp. 504–513.

[124] R.E. Patterson, *Human factors of stereoscopic 3d displays*, 1st ed., Springer-Verlag, London, 2015.

[125] K. Weigelt, J. Wiemeyer, *Depth perception and spatial presence experience in stereoscopic 3D sports broadcasts*, in: J.G. Verly, International Conference on 3D Imaging, 2012, Institute of Electrical and Electronics Engineers.

[126] M.M. Escobar, B. Juhnke, K. Hisley, D. Eliot, E. Winer, *Assessment of visual-spatial skills in medical context tasks when using monoscopic and stereoscopic visualization*, in: Proceedings of Medical Imaging 2013: Image Perception, Observer Performance, and Technology Assessment, Vol 8673, 2013, Society of Photo-Optical Instrumentation Engineers.

[127] R. Vlad, O. Nahorna, P. Ladret, A. Guerin, *The influence of the visualization task on the Simulator Sickness symptoms-a comparative SSQ study on 3DTV and 3D immersive glasses*, in: J. Watson (Ed.), 3DTV-Conference: The True Vision-Capture, Transmission and Dispaly of 3D Video, 2013, Institute of Electrical and Electronics Engineers, pp. 1–4.

[128] K.K. Bhagat, W.-K. Liou, C.-Y. *Chang, A cost-effective interactive 3D virtual reality system applied to military live firing training*, Virtual Reality 20:2 (2016) 127–140.

[129] N. Robitaille, P.L. Jackson, L.J. Hébert, C. Mercier, L.J. Bouyer, S. Fecteau, C.L. Richards, B.J. McFadyen, *A Virtual Reality avatar interaction (VRai) platform to assess residual executive dysfunction in active military personnel with previous mild traumatic brain injury: proof of concept*, Disability and Rehabilitation: Assistive Technology 12:7 (2016) 758–764.

[130] G. Gahm, G. Reger, M.V. Ingram, M. Reger, A. Rizzo, *A Multisite, Randomized Clinical Trial of Virtual Reality and Prolonged Exposure Therapy for Active Duty Soldiers with PTSD*, Maryland, 2015.

[131] D. Lazzari, G. Raimondo, *Digital Mock-Up in Support of Space Station Elements Integration and Tests*, in: B. Kaldeich-Schürmann (Ed.), Proceedings of 4th International Symposium on Environmental Testing for Space Programmes, 2001, European Space Agency, pp. 73–78.

[132] G. Sakas, U. Bockholt (Eds.), *Anwendungen der Virtuellen und Erweiterten Realität in der Medizin*, in: G. Horton, B. Preim, S. Schlechtweg, T. Schulze (Eds.), Proceedings der Tagung of Simulation und Visualisierung, 2005, SCS European Publishing House.

[133] R.M. Satava, *Medical applications of virtual reality*, Journal of Medical Systems 19:3 (1995) 275–280.

[134] A. Baldominos, Y. Saez, C. G. del Pozo, *An approach to physical rehabilitation using state-of-the-art virtual reality and motion tracking technologies*, Procedia Computer Science 64 (2015) 10–16.

[135] A.A. Rizzo, M. Schultheis, K.A. Kerns, C. Mateer, *Analysis of assets for virtual reality applications in neuropsychology*, Neuropsychological Rehabilitation 14:1/2 (2004) 207–240.

[136] J. Cukor, M. Gerardi, S. Alley, C. Reist, M. Roy, B.O. Rothbaum, J. Difede, A. Rizzo, *Virtual Reality Exposure Therapy for Combat-Related PTSD*, in: E. Ritchie (Eds.), Posttraumatic Stress Disorder and Related Diseases in Combat Veterans, Springer-Verlag, Cham, 2015, pp. 69–83.

[137] M.J. Smith, L. Boteler Humm, M.F. Fleming, N. Jordan, M.A. Wright, E.J. Ginger, K. Wright, D. Olsen, M.D. Bell, *Virtual reality job interview training for veterans with posttraumatic stress disorder*, Journal of Vocational Rehabilitation 42:3 (2015) 271–279.

[138] S.-n. Yang, T. Schlieski, B. Selmins, S.C. Cooper, R.A. Doherty, P.J. Corriveau, J.E. Sheedy, *Stereoscopic Viewing and Reported Perceived Immersion and Symptoms*, Optometry and Vision Science 89:7 (2012) 1068–1080.

[139] D. Wagner, T. Pintaric, F. Ledermann, D. Schmalstieg (Eds.), *Towards massively multi-user augmented reality on handheld devices*, in: H.W. Gellersen, R. Want, A. Schmidt (Eds) Pervasive Computing. Lecture Notes in Computer Science, Vol 3468. Springer-Verlag, Berlin Heidelberg, pp. 208–219.

[140] G. Kurillo, M. Forte, R. Bajcsy (Eds.), *Teleimmersive 3D collaborative environment for cyberarchaeology*, in: Computer Society Conference on Computer Vision and Pattern Recognition – Workshops, 2010, Institute of Electrical and Electronics Engineers, pp. 23–28.

[141] P. Astheimer, W. Felger, *Virtuelle Realität in der Architektur*, Bauinformatik 4:2 (1993) pp. 54–**58** .

[142] S.-c. Fan, F.-x. Wang, *Application of Virtual Reality in Tourism Management Professional Teaching*, in: J. Luo (Ed.), Soft Computing in Information Communication Technology, Vol 161, Springer-Verlag, Berlin Heidelberg, 2012, pp. 385–389.

[143] J.M. Varela, J.M. Rodrigues, C.G. Soares, *3D simulation of ship motions to support the planning of rescue operations on damaged ships*, Procedia Computer Science 51 (2015) 2397–2405.

[144] K.-P. Park, S.-H. Ham, S. Ha, *Validation of advanced evacuation analysis on passenger ships using experimental scenario and data of full-scale evacuation*, Computers in Industry 71 (2015) 103–115.

[145] R. Pérez Fernández, V. Alonso, *Virtual Reality in a shipbuilding environment*, Advances in Engineering Software 81 (2015) 30–40.

[146] J. Vora, S. Nair, A.K. Gramopadhye, A.T. Duchowski, B.J. Melloy, B. Kanki, *Using virtual reality technology for aircraft visual inspection training: presence and comparison studies*, Applied Ergonomics 33:6 (2002) 559–570.

[147] M. Durstewitz, B. Kiefner, R. Kueke, H. Putkonen, P. Repo, T. Tuikka, *Virtual collaboration environment for aircraft design*, in: A.D. Williams (Ed.), Proceedings of Sixth Conference on Information Visualisation, 2002, Institute of Electrical and Electronics Engineers, pp. 502–510.

[148] J. Xue, G. Zhao, *Interactive Rendering and Modification of Massive Aircraft CAD Models in Immersive Environment*, Computer-Aided Design and Applications 12:4 (2015) 393–402.

[149] A. Marzano, I. Friel, J.A. Erkoyuncu, S. Court, *Design of a Virtual Reality Framework for Maintainability and Assemblability Test of Complex Systems*, Procedia CIRP 37 (2015) 242–247.

[150] W. Schreiber, P. Zimmermann, *Virtuelle Techniken im industriellen Umfeld: Das AVILUS-Projekt Technologien und Anwendungen mit CD-ROM*, Springer-Verlag, Berlin Heidelberg, 2012.

[151] G. Döllner, P. Kellner, O. Tegel, *Digital Mock-up and Rapid Prototyping in Automotive Product Development*, Journal of Integrated Design and Process Design 4:1 (2000) 55–66.

[152] B. Rooks, *A shorter product development time with digital mock-up*, Assembly Automation 18:1 (1998) 34–38.

[153] F. Dai, W. Felger, T. Fruhauf, M. Gobel, D. Reiners, G. Zachmann, *Virtual prototyping examples for automotive industries*, in: J. Breit (Ed.), Virtual Reality World, 1996, Computerwoche Verlag.

[154] X. Wang, J.J.-H. Tsai, *Collaborative design in virtual environments*, Vol 48, Springer-Verlag., Netherlands, 2011.

[155] H. Salzmann, *Collaboration in Co-located Automotive Applications*, phD Thesis, 2010.

[156] M. Bordegoni, G. Caruso, *Mixed reality distributed platform for collaborative design review of automotive interiors*, Virtual and Physical Prototyping 7:4 (2012) 243–259.

[157] A. Gomes de Sá, G. Zachmann, *Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes*, Computer and Graphics 23:3 (1999) 389–403.

[158] S.K. Ong, A.Y.C. Nee, *Virtual and augmented reality applications in manufacturing*, 1st ed., Springer-Verlag, London, 2004.

[159] R. Stark, Beckmann-Dobrev B., Schulze E., Adenauer J., Israel J., Smart Hybrid Prototyping zur multimodalen Erlebbarkeit virtueller Prototypen innerhalb der Produktentstehung, in: A- Lichtenstein, Der Mensch im Mittelpunkt technischer Systeme. 8. Berliner Werkstatt Mensch-Maschine-Systeme, 2009, VDI-Verlag.

[160] T. Alt, W. Schreiber, W. Wohlgemuth, P. Zimmermann, *Das Verbundprojekt AVILUS*, in: W. Schreiber, P. Zimmermann (Eds.), Virtuelle Techniken im industriellen Umfeld: Das AVILUS-Projekt Technologien und Anwendungen mit CD-ROM, Springer-Verlag, Berlin Heidelberg, 2012, pp. 4–18.

[161] M. Schenk, M. Schumann, *Angewandte Virtuelle Techniken im Produktentstehungsprozess: AVILUSplus*, 1st ed., Springer Vieweg Verlag, Berlin Heidelberg, 2016.

[162] W. Friedrich, ARVIKA: *Augmented Reality für Entwicklung, Produktion und Service*, Publicis Corporate Publishing, Erlangen, 2004.

[163] F. Davide, P. Loreti, M. Lunghi, G. Riva, F. Vatalaro, *Communications through Virtual Technologies*, in: E. Gregori, G. Anastasi, S. Basagni (Eds) Advanced Lectures on Networking. Lecture Notes in Computer Science, vol 2497 Springer-Verlag, Berlin Heidelberg, 2002, pp. 124–157.

[164] G. Riva, *Virtual Reality as Communication Tool: A Sociocognitive Analysis*, Presence: Teleoperators and Virtual Environments 8:4 (1999) 462–468.

[165] F. Biocca, *Communication Within Virtual Reality: Creating a Space for Research*, Journal of Communication 42:4 (1992) 5–22.

[166] J. Bates, *Virtual Reality, Art, and Entertainment*, Presence: Teleoperators and Virtual Environments 1:1 (1992) 133–138.

[167] N. Döring, *Sozialpsychologie des Internet: Die Bedeutung des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen*, 2nd ed., Hogrefe Verlag, Göttingen, 2003.

[168] K. Beck, *Kommunikationswissenschaft*, 2nd ed., UTB Verlag, Stuttgart, 2016.

[169] G. Kurillo, R. Bajcsy, *3D teleimmersion for collaboration and interaction of geographically distributed users*, Virtual Reality 17:1 (2013) 29–43.

[170] M. Willert, S. Ohl, A. Lehmann, O.G. Staadt (Eds.), The Extended Window Metaphor for Large High-Resolution Displays, 2010.

[171] T. Islam, S. Ohl, O. Staadt, Multi-Camera Acquisition and Placement Strategy for Displaying High-Resolution Images for Telepresence Systems, in: M. Chover, A. Augusto de Sousa (Eds.), Eurographics 2013 - Posters, 2013, The Eurographics Association.

[172] J. Simard, M. Ammi, *Haptic interpersonal communication: gesture coordination for collaborative virtual assembly tasks*, Virtual Reality 16:3 2010 1–14.

[173] I.-H. Song, S.-C. Chung, *Synthesis of the digital mock-up system for heterogeneous CAD assembly*, Computers in Industry 60:5 (2009) 285–295.

[174] S. Kim, D. Weissmann, *Middleware-based integration of multiple CAD and PDM systems into virtual reality environment*, Computer-Aided Design & Applications 3:5 (2006) 547–556.

[175] P. Schneider, C. Clauß, A. Schneider, A. Stork, T. Bruder, T. Farkas, *Towards more insight with functional digital mockup*, in: K.W. Seibert (Ed.), Proceedings of the 4th EASC 2009 European Automotive Simulation Conference, 2009, ANSYS.

[176] M. Nybacka, T. Karlsson, T. Larsson, *Vehicle validation visualization*, in: Proceedings of Virtual Concept, 2006 Springer-Verlag.

[177] S. Fukudab, Z. Luliü, J. Stjepandiüa, *FDMU-Functional Spatial Experience beyond DMU?*, in C. Bil, J. Mo, J. Stjepandić (Eds.) 20th ISPE International Conference on Concurrent Engineering, 2013, iosPress, pp. 431–440.

[178] T. Farkas, A. Hinnerichs, C. Neumann, *An Interdisciplinary Approach to Functional Prototyping for Mechatronic Systems using Multi-Domain Simulation with Model-Based Debugging*, in: C.D. Zinn (Ed.), Proceedings of the International Multi-Conference on Engineering and Technological Innovation, Vol 2, 2008, nternational Institute of Informatics and Systemics.

[179] M. Sedlmair, W. Hintermaier, K. Stocker, T. Buring, A. Butz, *A dual-view visualization of in-car communication processes*, in: b. Werner (Ed.,) 12th International Conference Information Visualisation, 2008, Institute of Electrical and Electronics Engineers, pp. 157–162.

[180] M. Sedlmair, C. Bernhold, D. Herrscher, S. Boring, A. Butz, Mostvis: *An interactive visualization supporting automotive engineers in most catalog exploration*, in: S. Ceballos (Ed.), 13th International Conference Information Visualisation, 2009, Institute of Electrical and Electronics Engineers, pp. 173–182.

[181] M. Tonnis, R. Lindl, L. Walchshausl, G. Klinker, *Visualization of Spatial Sensor Data in the Context of Automotive Environment Perception Systems*, in: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, Institute of Electrical and Electronics Engineers, pp. 1–9.

[182] Stichting Blender Foundation, Blender.org, available at https://www.blender.org/ (accessed on June 9, 2017).

[183] M. Barranco, J. Proenza, G. Rodriguez-Navas, L. Almeida, *An active star topology for improving fault confinement in CAN networks*, IEEE Transactions on Industrial Informatics 2:2 (2006) 78–85.

[184] A.M. Barry, *Perception Theory*, in: K. Smith, S. Moriarty, G. Barbatsis, K. Kenney (Eds.), Handbook of Visual Communication: Theory, Methods, and Media, 1st ed., Routledge, Mahwah, 2005, pp. 45–62.

[185] M. Cohrs, V. Kremer, S. Klimke, G. Zachmann, *Time-efficient and accurate spatial localization of automotive function architectures with function-oriented 3D visualization*, Computer-Aided Design and Applications 13:4 (2016) 519–529.

[186] P.W. Jordan, B. Thomas, Weerdmeester, B.A., McClelland, I.L. (Eds.), *Usability Evaluation in Industry*, Taylor & Francis Ltd, London, 1996.

[187] M. Hassenzahl, M. Burmester, F. Koller, *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität*, in: G. Szwillus, J. Ziegler (Eds.), Mensch & Computer 2003. Berichte des German Chapter of the ACM, Vieweg+Teubner Verlag, vol 57, 2003, Vieweg+Teuber Verlag, pp. 187–196.

[188] J. Nielsen, 10 *Usability Heuristics for User Interface Design*, available at https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed on September 3, 2016).

[189] J. Nielsen, *Enhancing the explanatory power of usability heuristics*, in: B. Adelson, S. Dumais, J. Olson (Eds.), Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1994, Association of Computing Machinery, pp. 152–158.

[190] A. Cockburn, B. McKenzie, *Evaluating spatial memory in two and three dimensions*, International Journal of Human-Computer Studies 61:3 (2004) 359–373.

[191] S. Diefenbach, M. Hassenzahl, *The dilemma of the hedonic - Appreciated, but hard to justify*, Interacting with Computers 23:5 (2011) 461–472.

[192] M. Tory, T. Moller, M.S. Atkins, A.E. Kirkpatrick, *Combining 2D and 3D views for orientation and relative position tasks*, in: E. Dykstra-Erickson, M. Tscheligi (Eds.), Proceedings of the SIGCHI conference on Human factors in computing systems, 2004, Association of Computing Machinery, pp. 73–80.

[193] M. Tory, A.E. Kirkpatrick, M.S. Atkins, T. Moller, *Visualization task performance with 2D, 3D, and combination displays*, 1st ed., IEEE Transaction on Visualization and Computer Graphics 12:1 (2006) 2–13.

[194] A. Faath, R. Anderl, *Interdisciplinary and Consistent Use of a 3D CAD Model for CAx Education in Engineering Studies*, in: ASME 2016 International Mechanical Engineering Congress & Exposition, Vol 5, 2016, American Society of Mechanical Engineers.