systems

MDPI

Article

Methodology for Certification-Compliant Effect-Chain Modeling

Iris Gräßler [1,*], Dominik Wiechel [1], Anna-Sophie Koch [1], Tim Sturm [2] and Thomas Markfelder [2]

[1] Chair for Product Creation, Heinz Nixdorf Institute, Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany; dominik.wiechel@hni.upb.de (D.W.); anna.sophie.koch@hni.upb.de (A.-S.K.)
[2] 3DSE Management Consultants GmbH, Seidlstraße 18a, 80335 Munich, Germany; t.sturm@3dse.de (T.S.); t.markfelder@3dse.de (T.M.)
* Correspondence: iris.graessler@hni.upb.de

Abstract: The success of engineering complex technical systems is determined by meeting customer requirements and institutional regulations. One example relevant to the automobile industry is the United Nations Economic Commission of Europe (UN ECE), which specifies the homologation of automobile series and requires proof of traceability. The required traceability can be achieved by modeling system artifacts and their relations in a consistent, seamless model—an effect-chain model. Currently, no in-depth methodology exists to support engineers in developing certification-compliant effect-chain models. For this purpose, a new methodology for certification-compliant effect-chain modeling was developed, which includes extensions of an existing method, suitable models, and tools to support engineers in the modeling process. For evaluation purposes, applicability is proven based on the experience of more than 300 workshops at an automotive OEM and an automotive supplier. The following case example is chosen to demonstrate applicability: the development of a window lifter that has to meet the demands of UN ECE Regulations R156 and R21. Results indicate multiple benefits in supporting engineers with the certification-compliant modeling of effect chains. Three benefits are goal-oriented modeling to reduce the necessary modeling capacity, increasing model quality by applying information quality criteria, and the potential to reduce costs through automatable effect-chain analyses for technical changes. Further, companies in the automotive and other industries will benefit from increased modeling capabilities that can be used for architecture modeling and to comply with other regulations such as ASPICE or ISO 26262.

Keywords: effect chain; regulations; model-based systems engineering; traceability; RFLP; certification; automotive; systems engineering; impact analysis; SysML

check for updates

Citation: Gräßler, I.; Wiechel, D.; Koch, A.-S.; Sturm, T.; Markfelder, T. Methodology for Certification-Compliant Effect-Chain Modeling. *Systems* **2023**, *11*, 154. https://doi.org/10.3390/systems11030154

Academic Editors: Ed Pohl and Eric Specking

Received: 1 February 2023
Revised: 9 March 2023
Accepted: 15 March 2023
Published: 17 March 2023

1. Introduction

The complexity in the development of technical systems increases due to a high number of interdisciplinary system artifacts and relations between them. Complex technical systems from different domains are, for example, modern automobiles, medical patient systems, computers, mobile devices, and wearables [1]. To master the increasing complexity and resulting development effort, new approaches are necessary to build up development capabilities and digitalize the processes. Systems engineering is one possibility for developing complex technical systems in a structured, multidisciplinary way. [2] A promising approach to handle the complexity is to model the system artifacts and their relationships along the development phases. The artifacts include development, organizational, and environmental artifacts, which must be connected [3] (see Figure 1). Besides the complexity, the need for safety and security increases, which results in domain-specific regulations, norms, and standards. These regulatory documents (hereafter summarized as regulations) restrict the engineering of the systems [4] and have to be certified before entering the market [5]. The resulting impact is additional requirements that must be incorporated into the engineering process and time effort spent to audit the system. Therefore, it is a critical competitive factor to be able to certify and release systems according to the current regulative constraints. In general, these regulations require the creation of traceability between engineering artifacts and the definition of a corresponding traceability procedure [6]. Traceability is "the degree to which a relationship can be established between two or more products of the development process" [7] (p. 78). Examples from the automotive sector are the UN ECE regulations for homologation in European markets and requirements for certification according to ASPICE (Automotive Software Process Improvement and Capability Determination) [8].

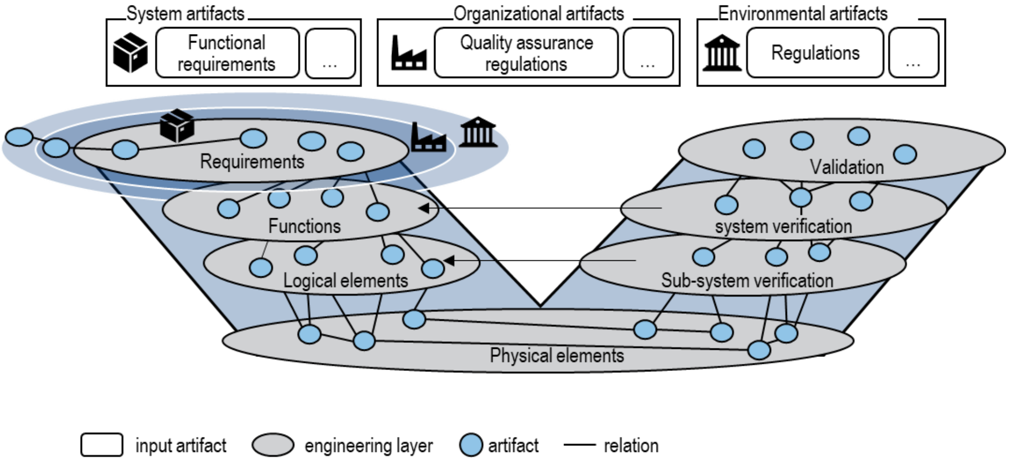


Figure 1. Variety of artifacts within the development of complex technical systems.

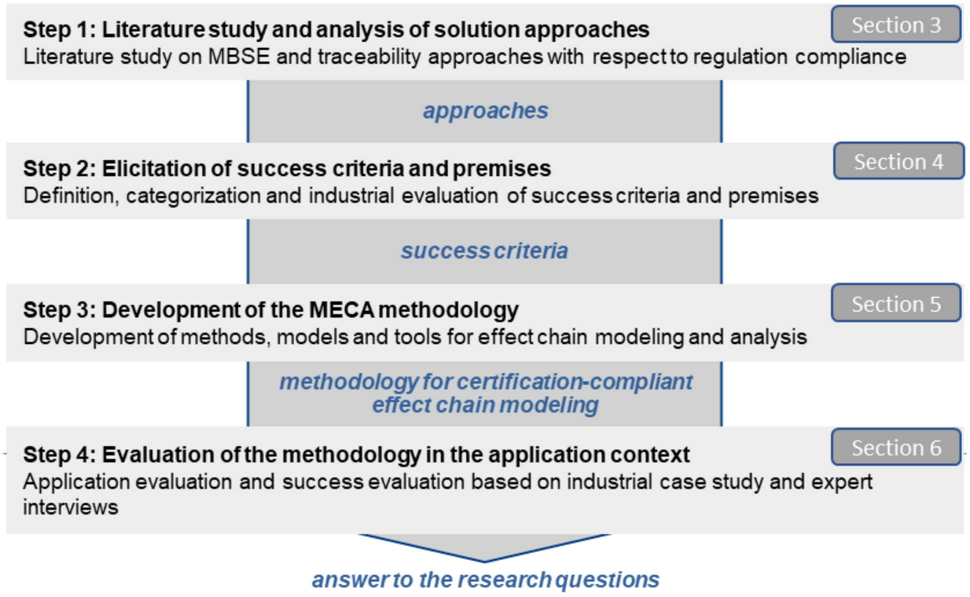
According to the study of NAIR ET AL. [9], just 5.7% of traceability research focuses on regulatory compliance using traceability models [10]. Traceability can be modeled as a separate traceability model or as part of a system model, which is the central artifact in model-based systems engineering (MBSE). Separate traceability models are created only to fulfill a regulation, which creates additional modeling effort detached from further development activities. Therefore, the use of MBSE modeling languages, tools, and methods supports developers in the efficient development of complex systems and certification in regulated environments [11]. Currently, no criteria exist to evaluate if the application of these approaches will be successful. In general, existing approaches are specified to individual regulations and do not provide additional assistance for engineers during the modeling process. Therefore, currently, no generic methodology exists which supports engineers in modeling certification-compliant system models and can be tailored to domain-specific regulation needs.

In the paper at hand, the authors propose a methodology for the certification-compliant modeling of effect chains, including methods, models, and tools [12,13]. Based on lessons learned from more than 300 workshops with engineers of an automotive OEM and an automotive supplier as well as the insights from three interviews with industrial modeling experts, the following research questions will be answered:

RQ1: Which success criteria need to be fulfilled by a certification-compliant modeling methodology?
RQ2: Which elements have to be included in a certification-compliant modeling methodology to fulfill the success criteria?
RQ3: How can a methodology be tailored to meet the needs of different regulations?

2. Scientific Approach

The scientific approach is based on the application-oriented research approach of Ulrich [14]. Additional aspects of the Design Research Methodology [15] are integrated, for example, the differentiation of success evaluation and application evaluation. As shown in Figure 2, the research is divided into four steps. In step one, a literature study focuses on methods to develop traceability models and MBSE system models to fulfill regulations. In addition to identifying existing solution approaches, relevant regulations that require traceability are identified. The literature study was conducted using the IEEE, Web of Science, and Scopus databases. Regulations, standards, and norms for the automotive, aerospace, healthcare, and railway domains are identified in the ISO Research Library and the VDI eLibrary. From the research results, seventeen articles were identified as appropriate for the context of this research. The identified modeling approaches are analyzed regarding their applicability for effect-chain modeling and analysis. Within step two, success criteria and premises for developing a methodology for certification-compliant effect-chain modeling are derived from the identified literature. These success criteria are checked and completed from a practical point of view within three expert interviews. The experts are a modeling expert from the German automotive industry and two modeling experts from a systems engineering consulting company. In step three, the new methodology (MECA) is developed, including methods, models, and tools for the certification-compliant modeling of effect chains. Tools are provided to assist engineers in "how" the method can be applied. Engineering models complete the methodology by defining "what" the necessary engineering activities are and by visualizing "where" the model elements are located within the development phases of a system. In addition to insights from the literature study, industrial insights and results of more than 300 performed workshops with experts from the automotive industry are included to develop, improve, and evaluate the methodology. In these industrial projects, more than 100 sub-systems were modeled according to a domain-specific regulation. In step four, the evaluation is conducted and divided into two parts: First, an application evaluation is performed by applying the methodology to an automotive sub-system—a window lifter. Second, a success evaluation is conducted based on the derived success criteria from step two.



Step 1: Literature study and analysis of solution approaches
Literature study on MBSE and traceability approaches with respect to regulation compliance
Section 3
approaches
Step 2: Elicitation of success criteria and premises
Definition, categorization and industrial evaluation of success criteria and premises
Section 4
success criteria
Step 3: Development of the MECA methodology
Development of methods, models and tools for effect chain modeling and analysis
Section 5
methodology for certification-compliant effect chain modeling
Step 4: Evaluation of the methodology in the application context
Application evaluation and success evaluation based on industrial case study and expert interviews
Section 6
answer to the research questions

Figure 2. Scientific approach.

In the following, Section 3 includes the literature study and analysis of solution approaches. In Section 4, the success criteria and premises are described. In Section 5, the development of the MECA methodology is described. The evaluation of the methodology in the application context is described in Section 6.

3. State of the Art

Within this section, the results of the literature study are presented. First, the literature study is conducted to identify approaches and compare their deficits and benefits. Then, based on these results, literature-based success criteria for a methodology for certification-compliant effect-chain modeling are derived in step two.

3.1. Regulations

Beginning with a look at the automotive industry, different regulations are relevant for engineering automotive systems. ISO 26262 is an international standard for ensuring the functional safety of road vehicles. The standard requires a high degree of formalization and traceability, for example, to avoid safety-critical inconsistencies between iterations in development and to allow interdisciplinary teams to work on a reference architecture [16]. In ASPICE [8], traceability and consistency are key concepts (cf. Annex D). Bidirectional traceability must be established between development artifacts such as system requirements, system architecture elements, or software components and associated test cases and results. The UN ECE defines regulations to cover requirements for the approval of the safety of vehicles. Moreover, from a software perspective, standards and process frameworks such as ISO/IEC 15504, capability maturity model integration (CMMI), or IEC 61508 suggest creating traceability between engineering artifacts to guarantee a high-quality system [17]. Within the aerospace industry, different regulations are defined, for example, the AS9100 [18], based on the international standard ISO 9001:2015. The demands of the AS9100 include the identification and traceability of outputs to ensure the conformity of systems and services. Exemplary demands are the identification of requirements and the assignment of a product to a batch of raw materials [18]. Additional standards for software systems in the aerospace industry are ISO12207 and DO-178B guidelines, which require trace links between requirements, design artifacts, and source codes [4]. In the health industry, standards such as IEC 62304 require traceability between system architecture elements and safety and risk aspects [19]. In railway systems, EN50128 is used to apply the standard IEC 61508 to the specific area of railway systems. Within EN50128, traceability is demanded between programmable electronic systems and their procedures and requirements affecting railway system safety-relevant areas [20]. A general guideline is the IEEE Guideline 24765:2010, “System and System Traceability”, which focuses on a standardized terminology that can be used in the context of all systems and software within the engineering field [21].

3.2. Traceability

The origins of the term traceability are based on requirements engineering [22]. From a requirements engineering perspective, traceability is defined as “the ability to describe and follow the life of a requirement, in both a forward and backward direction” [23] (p. 4). From an engineering perspective, traceability is “the degree to which relationship can be established between two or more products of the development process” [7] (p. 78). In this case, a product is an engineering artifact, e.g., a requirement or a logical element. Additionally, traceability can be described as the “ability to discover the history of every feature of a system” [24] (p. 1). From this perspective, traceability should allow the analysis of the design artifacts, starting from an initial point [25], and is demanded by standards, safety regulations, and maturity models [26]. Traceability is created by associating artifacts with trace links for a specific purpose [27]. Therefore, traceability focuses on the connectivity of information. Vertical traceability describes relations of dependent artifacts within a model, whereas horizontal traceability describes relations of dependent artifacts across models [28]. A description of artifacts and trace links is necessary for application and can be defined in a traceability information model (TIM). The TIM can be represented in different ways, for example, with the aid of modeling languages. In the TIM, model elements can be categorized into trace link classes, artifact classes, and trace path classes. [26]. The definition of the TIM includes, for example, the differentiation of link classes, which are modeled between two artifact classes [29]. One example is the definition of a trace link class called Verify to link the artifact class Test Case with the artifact class Requirement. Combining more than two artifact classes results in a trace path class.

The benefits of traceability are the support of communication [30] and the support of system verification and validation, impact analysis in engineering change management, and regulatory compliance [9]. Existing engineering knowledge can be persistent and reused in a new context [6]. Different studies underline the necessity of traceability for compliance in a regulated environment, for example, to meet maturity levels in automotive industrial applications [31]. There are still practical challenges that reduce the benefits of traceability in industry. One key challenge is that existing approaches give no guidance on identifying the necessary information and how to model them [6]. Other key challenges are the lack of integration across organizations and tools, missing goal-oriented approaches, and mapping to roles and responsibilities [30]. Additional challenges are the lack of knowledge and understanding about traceability [9] and the modeling of implicit knowledge of engineering, which is not yet formalized [32]. To increase the benefits of traceability for engineers, it is necessary to compensate the required effort and the resulting costs [29]. Therefore, the modeling of traceability information has to be goal-oriented [32], customizable, and extensible [33]. According to a study by Niar et al. [9], current research should focus on considering more artifacts, empirical evaluation, and additional tool application and adaption to support engineers.

3.3. Model-Based Systems Engineering Approaches

A significant enabler for successfully modeling traceability is a defined syntax and semantics [25]. MBSE focuses on creating system models using the syntax and semantics of modeling languages in a modeling tool with the aid of a modeling method [34]. The resulting system models include traceability information and serve as a central artifact in the systems engineering process. Therefore, MBSE approaches extend the scope of TIMs by using system models as engineering specifications. According to the International Council on Systems Engineering (INCOSE), MBSE "is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [35] (p. 15). Different modeling languages can be used to express the artifacts and relations of the model. Examples are the Unified Modeling Language (UML) [36] and the Systems Modeling Language (SysML) [37], which are proposed to model necessary traceability information [29]. In SysML, nine diagrams can be used to model a system's requirements, structure, and behavior. A SysML package diagram (pkg) is used to organize the system model. In addition, different types of requirements can be modeled in requirement diagrams (req). In the block definition diagram (bdd), various model elements and relationships can be modeled to express information about the system's structure [34]. Moreover, model elements and their properties are defined to specify the system within the bdd. The internal block diagram (ibd) is a complementary diagram and describes specific connections and flows between the structural parts. In the ibd, interfaces and static relations can be specified, for example, by differentiating energy, signal, and information flows. The activity diagrams (act), sequence diagrams (seq), and state machine diagrams (sd) express information about the dynamic behavior of a system [34]. Modeling approaches such as OOSEM [38] and SysMOD [39] use SysML to develop systems in a model-based and structured manner. The goal of these approaches is the structured and complete development of systems, not traceability and certification compliance. Besides SysML-based approaches, graph-based approaches support the modeling of traceability using graph databases and graph visualization techniques. Bajaj et al. [40] define a graph-based approach for modeling complex systems based on a digital blueprint called the Total System Model. The approach uses the digital blueprint to implement intra- and inter-relationships between different types of models.

The benefits of the MBSE of traceability are that existing system models can be used for both modeling the system to be developed and defining necessary traceability for the compliance of regulations. Other benefits are reduced development risk, improved quality and design integrity, and better knowledge transfer [38]. Additionally, MBSE approaches enable seamless and cost-free traceability and consistency between different design elements [16]. Therefore, MBSE can be used to show compliance in accordance with process-based standards, framework-based standards, or application-based standards [41]. According to DEAN ET AL. [11], model-based systems engineering approaches enable engineers to meet regulatory requirements by modeling a system's necessary confidence, traceability, and structure. Still, adoption challenges exist, for example, missing adaption strategies and guidelines for defining the purpose and scope of the modeling activities [42]. These adaption challenges and shortcomings underline that no generic methodology for certification-compliant modeling currently exists.

3.4. Identification of Existing Modeling Approaches

The identified solution approaches of the literature study are analyzed. Then, the analysis is divided according to the defined focus areas, resulting in "Certification-compliant traceability approaches" and "Certification-compliant MBSE approaches".

3.5. Certification-Compliant Traceability Approaches

Certification-compliant traceability approaches are used, for example, in the automotive industry to link stakeholder requirements to system model artifacts and to enable requirement management [16]. REMPEL AND MAEDER [26] describe a quality maturity model for establishing traceability to define and identify acceptable states and unacceptable deviations. REMPEL ET AL. [4] define an approach to derive certification-compliant traceability models for software-system-based discipline-specific guidelines. Available tools support information management and requirement traceability. IBM DOORS (Dynamic Object Oriented Requirement System) [6,43] is an IBM tool to trace and manage requirements using list formats and linkages to other model elements, e.g., in the system architecture in IBM Rhapsody. SLATE (System Level Automation Tool for Engineering) [32] is a tool-based approach using schemes to ensure traceability along a set of abstract blocks. Different authors extend the functionality of these software tools to support engineers in the fields of traceability. LAVAZZA AND VALETTO [44] extend IBM DOORS for the impact analysis of engineering change requests. The tool TOOR from PINHEIRO AND GOUGEN [45] is a supporting tool to record traceability between requirements and other artifacts. EMFTrace is a tool based on the eclipse modeling framework (EMF) which supports engineers in storing and connecting traceability information [46]. Additionally, HEGEDÜS ET AL. [47] developed the extension EMF-IncQuery which is used in a case study for the compliance of traceability of the certification standard DO-178b in the avionic domain. Another approach in the context of regulatory compliance is the tool READs, which supports the engineering process of the standard DOD-STD-2176A [48]. BARBOSA ET AL. define traceability information models in the context of compliance-relevant development in the connected health industry [19].

The findings of the literature study are that the certification-compliant modeling approaches are defined for a specific regulation, standard, or norm. Therefore, these approaches cannot be adapted to other regulations. Additionally, the applicability of these approaches is limited to the level of detail of the descriptions. The mentioned tools support engineers in the traceability of requirements to other elements but lack methodological support for modeling traceability information in a certification-compliant context. Therefore, no generic and applicable traceability approach or traceability tool exists to support engineers in the modeling of their domain-specific traceability demands for certification-compliant purposes.

3.6. Certification-Compliant MBSE Approaches

Different MBSE approaches for certification-compliant modeling exist, for example, in the domains of automotive, aviation, and critical infrastructure. WEILKIENS ET AL. [49] compare existing approaches for mapping safety-critical information in SysML system models. DAVID AND SHAWKY [50] define a SysML profile to be able to represent the artifacts to meet traceability aspects of ISO 26262 in a system model. DABRINGER ET AL. [51] define an approach for linking requirements with the system architecture and FMEAs. HILLENBRAND [52] describes how requirements of ISO 26262 can be considered during the modeling of system architectures, focusing on electrical and electronic systems to meet the requirements of the standards. GRÄSSLER ET AL. [3] focus on the certification-compliant modeling of effect chains to fulfill the UN ECE regulation for automobile series. Additional authors developed approaches for the aviation domain, for example, to fulfill the airworthiness certification [53], the airframe segment [54], or specific regulations such as CFR Part 25 [55]. SANNIER ET AL. [56] describe an approach to fulfill safety regulations and standards for power plant systems based on a traceability concept and model-driven techniques.

Overall, regulation-specific approaches exist for certification-compliant MBSE. Findings are that a generic and adaptable approach is only given by the MECA method, which does not include models and tools for the detailed application. Additionally, access to existing SysML profiles and regulation-specific adaptions is restricted. As with the traceability approaches, the level of detail is not sufficient to enable application in practice on the basis of the descriptions. The limitations are depicted in Figure 3.

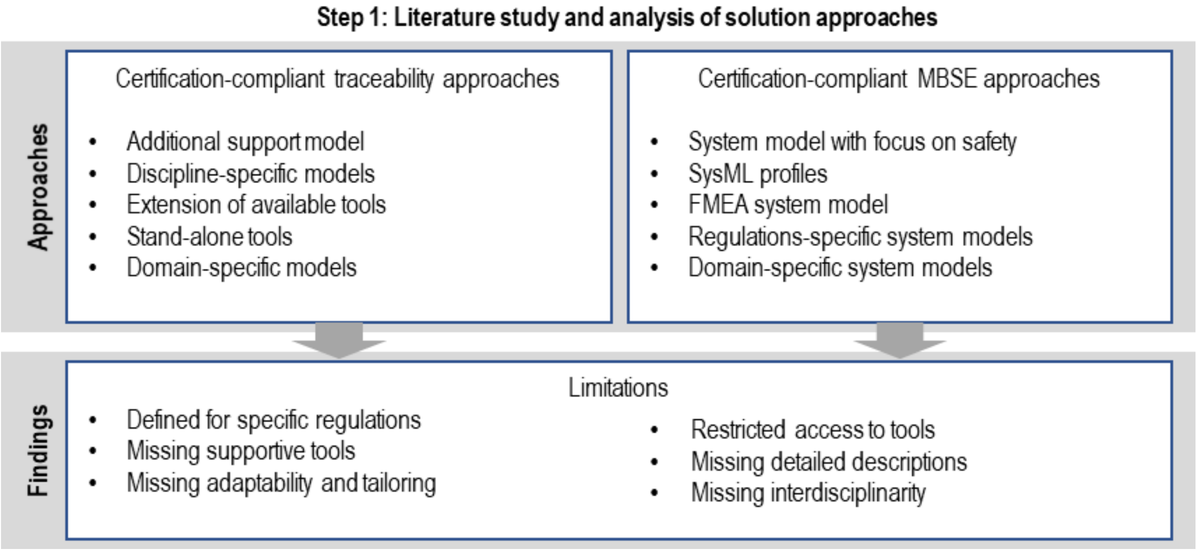

Step 1: Literature study and analysis of solution approaches
Approaches
Certification-compliant traceability approaches
• Additional support model
• Discipline-specific models
• Extension of available tools
• Stand-alone tools
• Domain-specific models
Certification-compliant MBSE approaches
• System model with focus on safety
• SysML profiles
• FMEA system model
• Regulations-specific system models
• Domain-specific system models
Findings
Limitations
• Defined for specific regulations
• Missing supportive tools
• Missing adaptability and tailoring
• Restricted access to tools
• Missing detailed descriptions
• Missing interdisciplinarity

Figure 3. Results of the literature study [3,4,19,26,44–56].

4. Success Criteria for the Methodology

In the next step, success criteria (SC) for the systematic development of a certification-compliant modeling methodology are compiled and divided into input, methodology, application, and output [57]. The SC are defined in two steps: (1) deriving SC from literature-based challenges and limitations of existing solution approaches and (2) evaluating the completeness and validity of the SC in interviews from a practical point of view with industrial modeling experts. For the second step, three interviews were performed with an expert in model-based system engineering working at a large German automotive OEM company and two experts from a German system engineering consulting. The feedback of the interview partners is used to ensure the validity and completeness of the SC. The interviews were performed in the form of semi-structured interviews. To ensure the completeness of the SC, the SC predefined from the literature were discussed. In addition, the experts were asked to give feedback on the relevance of the SC and their level of granularity. Exemplary feedback from the interview partners to complete the SC was that an SC has to be added, focusing on the processability of different data formats (SC-3). The results of the synthesis are 12 SC (see Table 1).

Table 1. Success criteria for the development of the methodology.

Input	**Application**
SC-1: Integrability of interdisciplinary artifacts	SC-7: Applicability independent of the regulations
SC-2: Processability of high number of artifacts	SC-8: Acceptable application effort
SC-3: Processability of different data formats	SC-9: Collaborative applicability
SC-4: Assurance of information quality	SC-10: Applicability without effect chain related knowledge
Output	**Methodology**
SC-5: Indication of resulting effects	SC-11: Ability to model interdisciplinary effects
SC-6: Sufficient accuracy of effects	SC-12: Goal-orientation of modeling steps

SC = Success Criteria.

With the aid of the interview partners, the success criteria were completed to answer RQ1. Exemplary adjustments based on the interviews were that the partners stated that SC-5 (indication of resulting effects) is the most important success criterion since it is the primary motivation for modeling certification-compliant effect chains. Additionally, the interview partners stated that applicability has to be independent of a specific regulation (SC-7) because different regulations have to be fulfilled in their domains. Two of the interview partners stated that besides success criteria, premises for a methodology are necessary as well. These premises must be fulfilled before a company's effect-chain modeling methodology is implemented. These premises are:

- P-1: Availability of required software;
- P-2: Availability of required information;
- P-3: Compatibility of existing IT infrastructure.

5. Methodology for Certification-Compliant Effect-Chain Modeling

To answer RQ2, the generic methodology for certification-compliant effect-chain modeling is defined (see Figure 4). In this case, an existing method from the literature [3] is chosen, extended, and detailed. All in all, the methodology supports engineers in modeling the necessary traceability information to a specific regulation within a system model. Furthermore, depending on the compliance-relevant needs, the method is adaptable to the necessary granularity of design representation, from a high-level architectural view down to a detailed design view.

Step 1: Goal definition of effect-chain modeling

In the first step, the goal of effect-chain modeling is defined. Beginning with the activity analyze system, the system of interest (SOI) and its system boundaries have to be clearly defined and differentiated from other systems within the system context [58]. For this specified system context, in the activity analyze regulations existing regulations from strategic planning and associated models (e.g., application scenarios) are identified and investigated. These regulations have to be analyzed to identify the required demands to fulfill the regulations. Conflicts with other regulations, standards, or specifications are avoided by extracting all text passages from regulations to identify and consolidate the regulatory demands. These text passages include statements for the certification and are formalized in the Traceability Information Need (TIN). One example passage is "trace software tests to software requirements". The TIN includes the required traceability artifact classes, traceability relationships, and traceability paths to fulfill the text passages. Therefore, it is the required information to fulfill the certification and develop the model. Hence, a specific modeling depth, the so-called modeling granularity, has to be chosen for each artifact. In the activity analyze granularity, the modeling depth of each traceability artifact class and relationship is specified. The granularity can differ from a generic description of an element as a simple node up to detailed descriptions of the class and its properties (part, value, references, etc.). As assistance for the engineer, the tool called "RFLPV handouts" [59] can be used to give guidance on how to define the granularity and how to elicit the necessary information. The RFLPV handouts include definitions, processes, control mechanisms, and enablers to identify the information of typical interdisciplinary engineering artifacts (requirements, functions, logical elements, physical elements, verification elements).

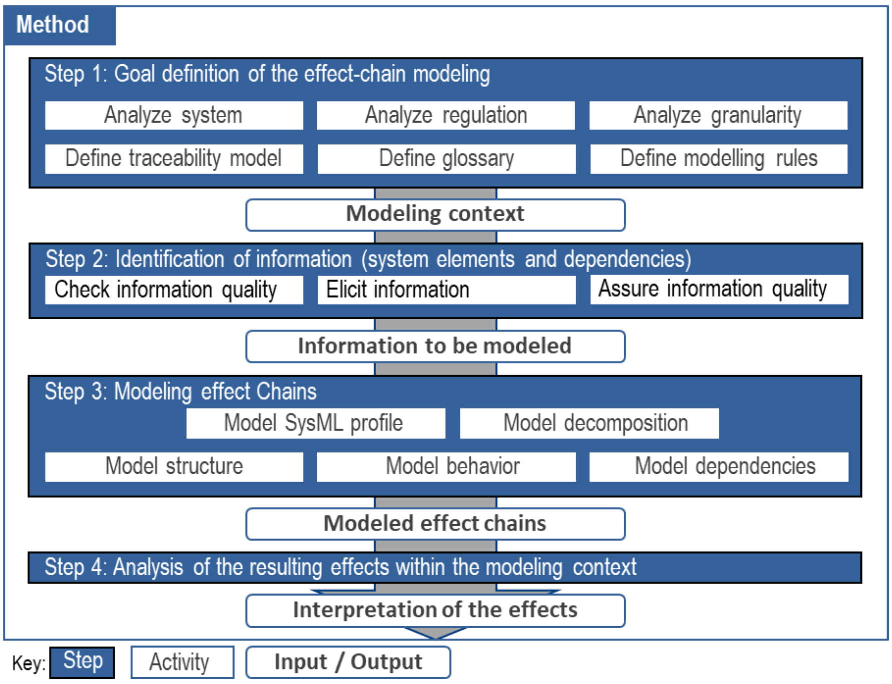

Method
Step 1: Goal definition of the effect-chain modeling
Analyze system
Analyze regulation
Analyze granularity
Define traceability model
Define glossary
Define modelling rules
Modeling context
Step 2: Identification of information (system elements and dependencies)
Check information quality
Elicit information
Assure information quality
Information to be modeled
Step 3: Modeling effect Chains
Model SysML profile
Model decomposition
Model structure
Model behavior
Model dependencies
Modeled effect chains
Step 4: Analysis of the resulting effects within the modeling context
Interpretation of the effects
Key:
Step
Activity
Input / Output

Figure 4. Overview of the MECA method (based on [3]).

Based on the analysis and the resulting TIN, the context-specific TIM is derived and formalized in the activity define traceability model. The TIM includes the necessary semantics, syntax, and terminology to verify the modeling in accordance with the TIN. For each identified artifact in the TIN, artifact classes and link classes are defined in the TIM. One permission for the definition of the TIM is to achieve the necessary traceability while minimizing the number of artifact classes and link classes within the model [29]. Within the TIM, the following questions should be answered [25]:

- What are the traceable artifacts that shall be represented?
- What are the relations and characteristics of the artifacts?
- Who are the actors and roles that are included in the modeling process?
- Which information granularity level shall be used to trace every engineering artifact?

Besides the definition of the TIM, a glossary and specific modeling rules are derived and captured in SysML diagrams. In the activity define glossary, the required terminology of the TIM is captured transparently, for example, definitions of artifact classes. Besides defining "what" the elements are, "how" the model has to be filled is also defined. Therefore, in the activity define modeling rules, a set of rules is derived to guide the participating modeling engineers during the collaboration [41]. This ruleset includes guidelines and best practices to develop a complete and consistent model. An exemplary rule is "each requirement has to be fulfilled by at least one test case". Besides that, tools are used for visualizing and mapping the TIM elements to the company-specific processes. Different roles and actors manage the information on the required TIM artifact classes and relationships within the company. Existing role models, such as the role model of MBSE [60], can be used as a starting point to define participants' activities. For each group of stakeholders, a viewpoint and specific views can be derived, which target the audience, imply necessary information, and are consistent with the defined need [41]. Additionally, the artifact classes and link classes of the TIM can be mapped to existing engineering models such as the well-known V-Model of the VDI/VDE 2206:2021 [61–64]. This step supports the transparency, visibility, and communication in the company and across their boundaries, for example, when presenting the approach to auditors of the regulation.

Based on the goal analysis, the definition of the required information, and the application of support elements, the first step results in the TIM and the modeling context, which is aligned with the goal of effect-chain modeling (see Figure 5). For the following steps, a new permission can be applied: every activity carried out in the modeling context must contribute to the goal [41].

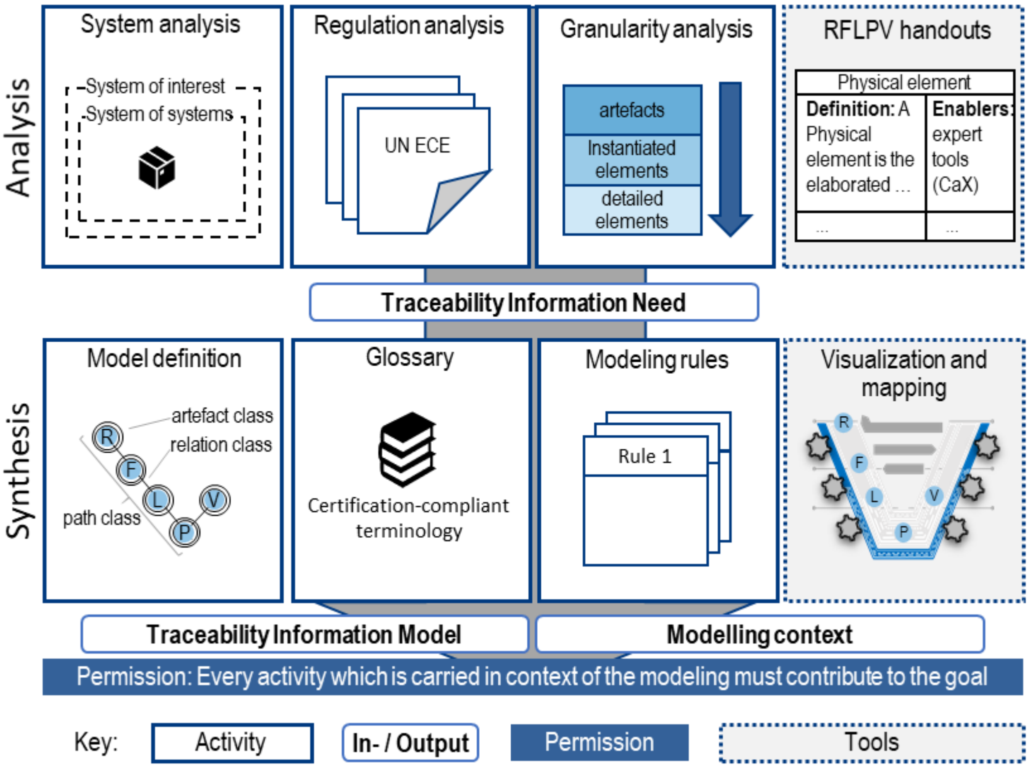


Figure 5. Goal definition of effect-chain modeling.

Step 2: Identification of information

In step 2, information to model the TIM is identified and consolidated. At this point, in a comprehensive industrial modeling project with a German automotive OEM, more than 150 of the 300 workshops were conducted to identify the necessary information. According to Gräßler et al. [3], the time effort of the identification of information exceeds the time effort of modeling. Therefore, identifying information is a critical step for a successful modeling approach. Based on the insights from the industrial projects, the identification of information is divided into information availability, information elicitation, and information quality assurance:

Check information availability

In the activity check information availability, the engineer checks if the required information to complete the TIM already exists in the company. Each required trace link artifact and trace link class is checked and documented. For documentation, collecting information about the source, title, subject, information type, and relations is suggested. The output of this step is a set of information for modeling the effect chain, the documentation of the available information, and a set of missing information. For this set of missing information, information elicitation has to be executed.

Elicit information

In the activity elicit information, different methods for information elicitation can be used, for example, design structure matrixes (DSMs) [65], in-depth interviews [66], focus groups [67], questionnaires, or the analysis of documents and data. Each set of information has to be classified and described by the suggestions above to verify applicability as input for the model.

Assure information quality

Besides specifying the information, information quality (IQ) is proven in the activity assure information quality. Problems with information quality can occur because of incorrect links, a change in context, and changes in information entities and references [68]. Therefore, a set of information quality criteria is derived from existing literature on information technology. The generic set can be extended to context-specific needs, for example, using IQ dimensions of the framework for information quality criteria [68]. Based on existing literature [69–71], the following IQ criteria for the input information for certification-compliant effect-chain modeling are derived:

Faultlessness: Information is error-free if it matches reality.

Accessibility: Information is accessible if it can be retrieved by simple procedures and accessed directly by the user.

Timeliness: Information is up-to-date if it reflects the actual properties of the described object in a timely manner.

Completeness: Information is complete if it is not missing and is available at the defined time in the respective process steps.

Based on the information elicitation and the information quality assurance, missing information can be elicited, and all input information for effect-chain modeling is derived from filling the effect-chain model (see Figure 6).

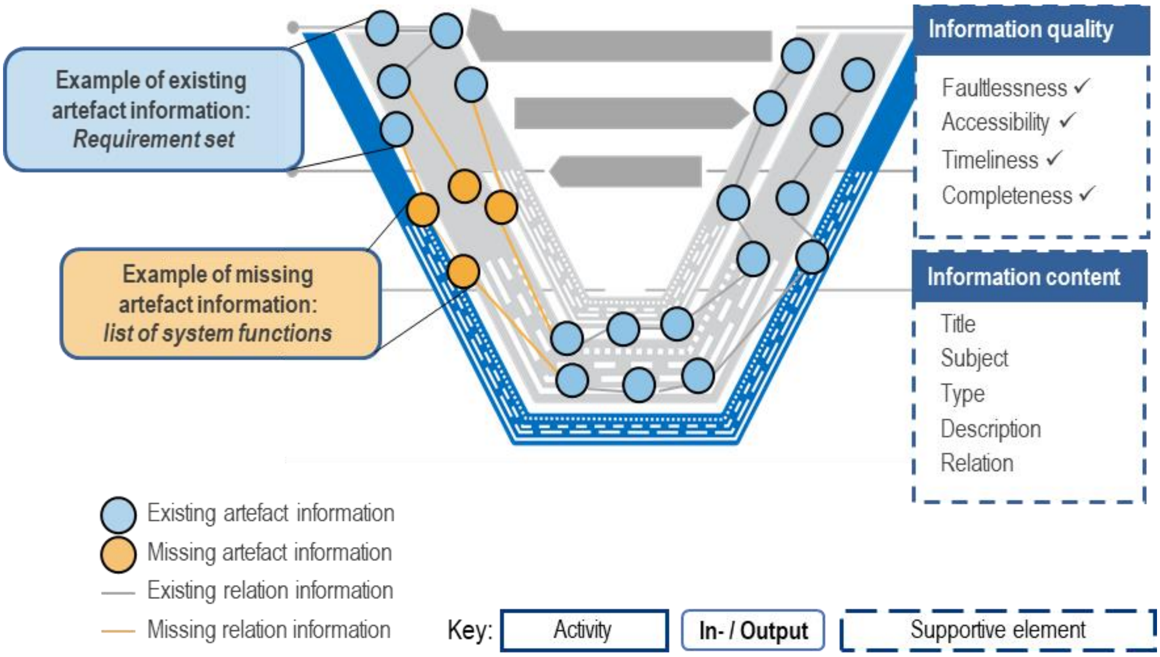

Example of existing artefact information: Requirement set
Example of missing artefact information: list of system functions
Information quality
Faultlessness ✓
Accessibility ✓
Timeliness ✓
Completeness ✓
Information content
Title
Subject
Type
Description
Relation
Existing artefact information
Missing artefact information
Existing relation information
Missing relation information
Key:
Activity
In- / Output
Supportive element

Figure 6. Identification of information along the V-Model.

Step 3: Modeling effect chains

In step 3, the certification-compliant effect-chain model, which represents the elicited information (step 2) and fulfills the defined goal (step 1), is developed in five interdependent activities (see Figure 7). At this point, the authors suggest the usage of SysML as a modeling language due to its expandability and practical relevance.

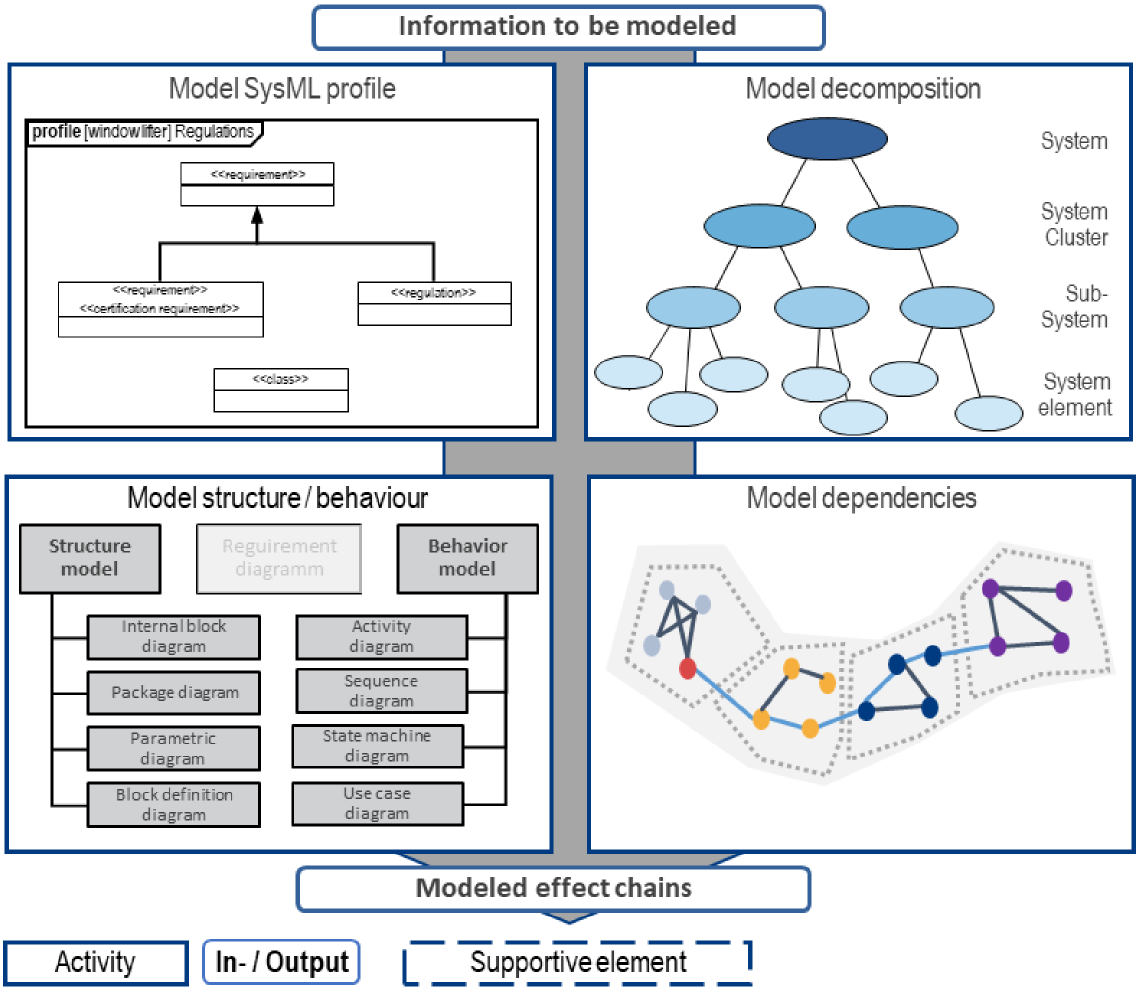


Figure 7. Step 3 Modeling effect chains.

Model SysML profile

Depending on the derived information from step 1, a SysML profile is defined to extend the meta model following the TIM. Each existing <<metaclass>> of the meta model of the modeling language can be extended within a profile diagram. For each model element and relationship, which is currently not part of the meta model, the appropriate <<metaclass>> is extended by new <<stereotypes>>, which can be named in accordance with artifact classes and link classes in the glossary. Besides the customization of typical model elements such as <<block>>, specific model elements can be extended to support the model's applicability. One example is the customization of <<viewpoints>> and <<views>> for stakeholders that have an interest or are participating in the model. These stakeholders can be internal and external persons participating in the modeling.

Model decomposition

After defining the SysML profile, the system of interest (SOI) is decomposed. Decomposition is defined as the step-by-step subdivision of a system, which can be divided into successive levels, for example, overall systems, sub-systems, and system elements [58,61].

In the V-Model, the decomposition on the left wing includes the decomposition of the system elements into sub-systems and the overall system [58]. In contrast to that, the right wing of the V-Model includes the integration of the system elements as well as the validating and verifying properties. [72] Depending on the point of view and the focused model elements, the decomposition includes different artifacts of the system's architecture, such as requirements, functions, logical elements, and physical elements [73]. These elements can be decomposed hierarchically to subdivide the system into several levels [74]. A hierarchical subdivision of the artifacts helps identify the dependencies between homogenous artifacts, for example, between stakeholder requirements and derived system requirements. Based on the hierarchical decomposition, the system model is organized in a pgk. The segmentation of the uniform packages depends on the TIN. The result of the decomposition is an unfilled package structure, which is represented in the containment tree of the modeling tool.

Model structure

The structure of a system is included in the system boundary and defines the system parts and their corresponding relationships and interactions [58]. Different representations help model interrelated perspectives of the structure. In engineering, the black box and white box method help understand the structure of a system [75]. A black box represents an external view of a system, whereas a white box represents the internal view [58]. In SysML, these concepts are included in the block definition diagram (bbd) and the internal block diagram (ibd) [37].

Model behavior

Besides the structure model, SysML diagrams for modeling the dynamic behavior of the system and its elements exist. The behavior is applied on specific levels of the predefined structure and includes dynamic relations to express ordered procedures and information sequences. Depending on the TIM, the system's behavior is identified, and diagrams representing the behavior are selected. Examples are a set of actions in an activity diagram, the expression of messages between system parts in a sequence diagram, or the differentiation of system statuses in a state machine diagram. The behavior model results from a set of diagrams describing systems' dynamic behavior on different levels.

Model dependencies

In the end, the relations between the model artifacts are completed. It is suggested to differentiate between different trace link classes to increase the understanding of each relation's purpose and to increase opportunities for more dedicated analysis of the model [29]. It is also suggested to link the elements, starting from the beginning of the engineering process, from the requirements down to the functions, logical elements, physical elements, and verification (RFLPV). In SysML, different relations are predefined, for example, <<allocate>>, <<satisfy>>, <<trace>>, or <<verify>>. These predefined relations are extended by the SysML Profile (Step 3.1) to explicitly model the defined trace link classes from the TIM between the system artifacts. This step results in a complete set of relations between relevant system artifacts in harmony with the TIM.

Since modeling implies that only a reduced amount of information is represented [76], the use of control questions is suggested to ensure the quality of the model. Besides existing control questions for MBSE [77], the following control questions are defined (according to [38]):

- Is the model's scope sufficient to meet its intended use?
- Is the model complete relative to its scope?
- Is the model consistent and understandable?

Step 4: Analysis of the effects within the modeling context

The resulting system model from step 3 is used to analyze effects, for example, affected elements from regulation or impacts of engineering changes [78]. Different analysis types can be used to analyze the effect chains, for example, validating the correctness of modeled relations, identifying gaps in coverage, counting existing relations for reports, or evaluating the impacts of engineering changes [29]. Additionally, specific analysis can be conducted, for example, analyzing elements in different system variants affected by a technical engineering change request or a software update [79]. The analysis can be divided into expert-based, semi-automated, and automated. In expert-based analysis, involved persons use the model to ensure certification compliance. Besides that, developers can also use tools and algorithms to infer queries that analyze the effect-chain model semi-automatically. In addition to semi-automated approaches, artificial intelligence (AI) offers the potential to learn from statements and infer further statements automatically. AI approaches can also assist in identifying relations [80] or assessing the change risk [81].

The developed MECA methodology comprises an extended four-step method, a representation along the V-model, and several tools (see Figure 8).

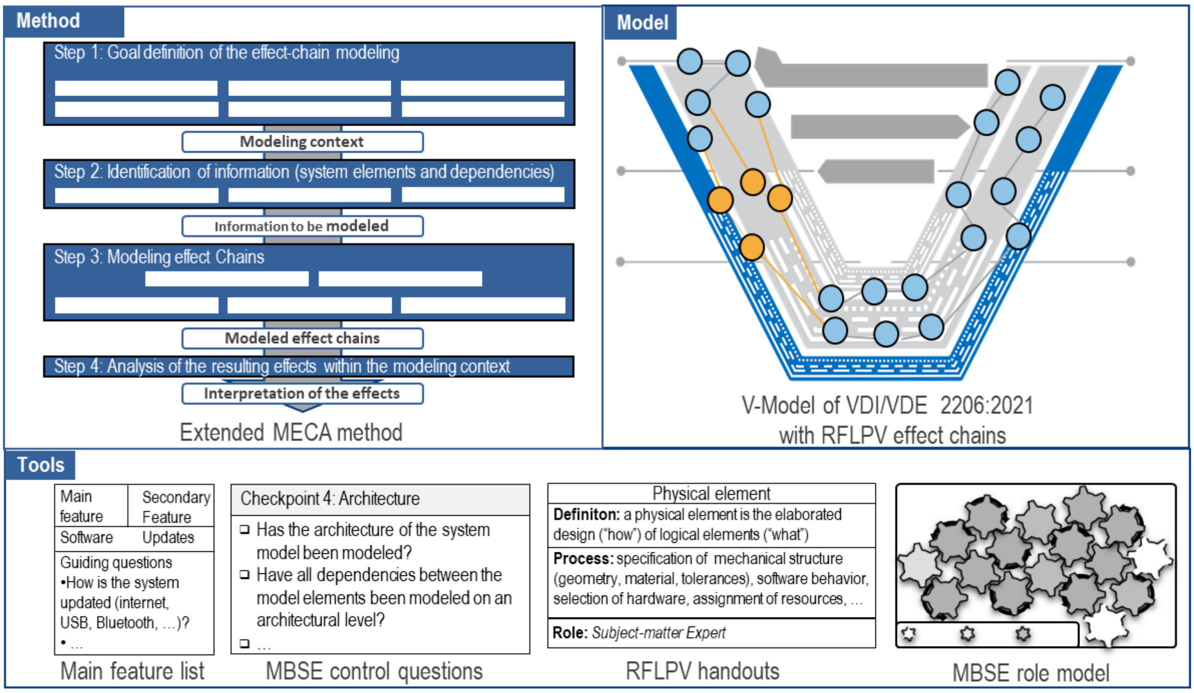

Method
Step 1: Goal definition of the effect-chain modeling
Modeling context
Step 2: Identification of information (system elements and dependencies)
Information to be modeled
Step 3: Modeling effect Chains
Modeled effect chains
Step 4: Analysis of the resulting effects within the modeling context
Interpretation of the effects
Extended MECA method
Model
V-Model of VDI/VDE 2206:2021
with RFLPV effect chains
Tools
Main feature
Secondary Feature
Software
Updates
Guiding questions
•How is the system updated (internet, USB, Bluetooth, …)?
• …
Main feature list
Checkpoint 4: Architecture
❑ Has the architecture of the system model been modeled?
❑ Have all dependencies between the model elements been modeled on an architectural level?
❑ …
MBSE control questions
Physical element
Definiton: a physical element is the elaborated design ("how") of logical elements ("what")
Process: specification of mechanical structure (geometry, material, tolerances), software behavior, selection of hardware, assignment of resources, …
Role: Subject-matter Expert
RFLPV handouts
MBSE role model

Figure 8. Overview of the MECA methodology [3,59,61,62,77].

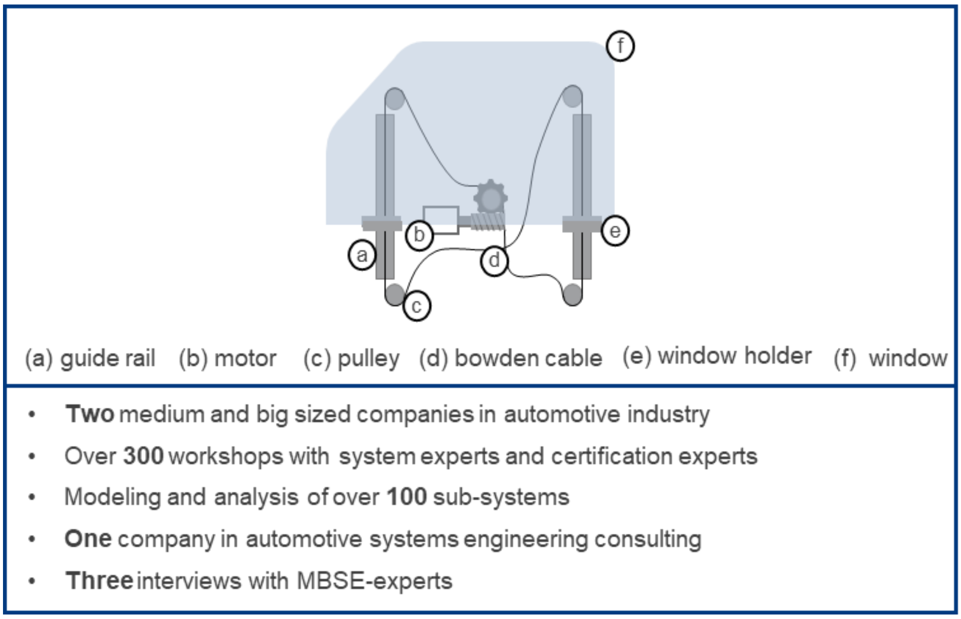
6. Evaluation

This section is divided into the application evaluation of the MECA methodology in a case example from the automotive domain and the successful evaluation of the methodology based on the derived success criteria.

6.1. *Evaluation of the Methodology in the Application Context*

Since the approval of the vehicles is decided based on the effect chains, correct and efficient application of the methodology has the highest priority, which is why great effort was put into the validation. The MECA method was created based on the experience of 300 workshops in a fourteen-month industry project with a German automotive OEM. As a result, the method was proved supportive but is missing a more detailed description, corresponding models, and tools to guide the modeling experts. Therefore, those experiences were used to develop the MECA methodology, which was applied in a medium-sized automotive supplier. The application of the MECA methodology is used to prove the applicability and transferability of the methodology to other companies. In these projects, more than 100 sub-systems of a vehicle have been analyzed and modeled. The workshops have been conducted with certification experts and systems engineers to identify the necessary information and validate the effect chains of the modeling process. In the following, one sub-system is used as a case example to demonstrate the application of the MECA methodology. Afterward, the success evaluation is conducted based on the success criteria (step 2).

For the case example, a window lifter is the system of interest. The window lifer is integrated into an automotive door system, which is integrated into the overall vehicle. The window lifter is used for opening and closing the windows of the vehicle, which can be conducted manually and automatically. The schematic sketch of the window lifter is shown in Figure 9.



- **Two** medium and big sized companies in automotive industry
- Over **300** workshops with system experts and certification experts
- Modeling and analysis of over **100** sub-systems
- **One** company in automotive systems engineering consulting
- **Three** interviews with MBSE-experts

Figure 9. Case example of a window lifter [82].

Step 1: Goal definition for effect-chain modeling

Analyze system: The window lifter is a sub-system of the system vehicle. In the automotive industry, function-oriented modeling focuses on customer demand [83]. The main functions of the window lifter are that the windows of the vehicle can be opened and closed automatically or manually by triggering a signal with different devices, such as the driver's voice or the manual button in the vehicle.

Analyze regulation: The regulations relevant for the window lifter are the UN-ECE regulation No. 156 (R 156) and UN ECE Regulation No. 21 (R 21), published by the United Nations Economic Commission for Europe (UN Regulation No. 156—Software update and software update management system | UNECE). The R 156 addresses the "approval of vehicles with regard to software update and software updates management system" [84]. The regulation allows an organization to make updates on a vehicle which are already approved in the type approval. By the regulation, a standardized evaluation of the update is required, including the identification of impacted elements and dependencies of the overall system. [84]. In R 21, it is defined that the window lifter has to include anti-trap protection. Therefore, if a clamping force of more than 100 newtons is detected, the window lifter has to open the window to ensure the safety of the passengers and reduce injuries due to clamping. [85]. Hence, it needs to be proven if a software update on the electric control units (ECUs) of the vehicle affects the functional behavior of the window lifter or other related sub-systems.

Analyze granularity: In the granularity analysis, each trace artifact class is investigated. To fulfill the demands of UN ECE regulations, each regulation is modeled as a separate entity. The functional structure and behavior are modeled by separating two levels: customer functions represent the necessary functions from a customer perspective, whereas system functions represent the behavior from a technical perspective. As a result, several system functions from different sub-systems have to be linked together to fulfill the custom functions. Additionally, the information flows between system functions need to be differentiated. The functional behavior is implemented on software and hardware elements. Therefore, system functions must be linked to the vehicle's sensors, actuators, and ECUs.

Based on the analysis, the TIN is defined. The analysis of the R 156 shows that traceability between software components that can be updated and elements that fulfill UN ECE regulations has to be defined. In consultation with the corresponding system engineers, customer and system functions build the bridge between regulations and system components. Therefore, the TIN includes engineering artifacts and dependencies to map regulation documents to customer functions and system functions, as well as their dependencies to executing hardware and software elements.

Define traceability model: A context-specific TIM is derived and formalized with the modeling language SysML based on the TIN. Therefore, artifact classes, link classes, and path classes are specified:

1. Artifact classes: Artifact classes are regulations, certification requirements, customer functions, system functions, hardware components, and software components which are modeled as stereotyped SysML <<Block>> elements.
2. Link classes: The link classes contain allocations of system functions to regulations and components as well as connectors between system functions and are modeled with the aid of the standard SysML model elements <<allocate>> and <<connector>> [37].
3. Path classes: The main path class is the linkage between regulations allocated to the functional behavior of system functions implemented on components and modeled with the aid of standard SysML diagrams [37] (pgk, req, bdd, ibd).

Define glossary: In order to create a uniform understanding for all participants, the following terminology is used in the context of modeling the effect chains:

- System function (SF) = A system function executes a customer function in interaction with other system functions.
- Customer function (CF) = A customer function is executed by a sub-system and represents a specific system that is recognizable by a customer. Every customer function is implemented by one or more system functions.

Define modeling rules: For assistance during the modeling process, the following modeling rules are determined:

1. Always refer to the system function of a system to model dependencies to other systems.
2. Map regulation to system function, not to customer functions.
3. For mapping relations between different systems, system functions are used.
4. Use bidirectional connectors without naming to connect system functions.
5. Define ports for each interface between system functions and name them according to the transmitted information.
6. Categorize the system functions of a customer function into the categories: input, processing, and output.

Visualization and mapping: As support during the application of the methodology, the participating roles, artifact classes, and artifact links are visualized by the company-specific product development task, represented by the V-Model (see Figure 10).

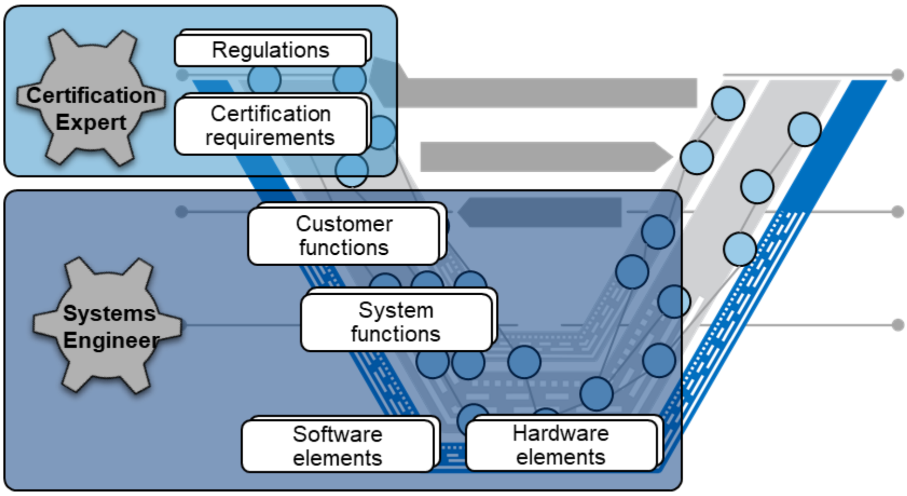


Figure 10. Mapping of roles and trace artifact classes at the V-Model.

Step 2: Information Elicitation

Check information availability: The available information is offered in Excel sheets. A total of 170 separated regulations are listed in a table. The components list includes all sensors, actuators, and ECUs that are installed in the vehicle series. Regarding the system architecture, a list of all customer functions and system functions of the vehicle series is provided. Examples of the customer and system functions are shown in Figure 11.

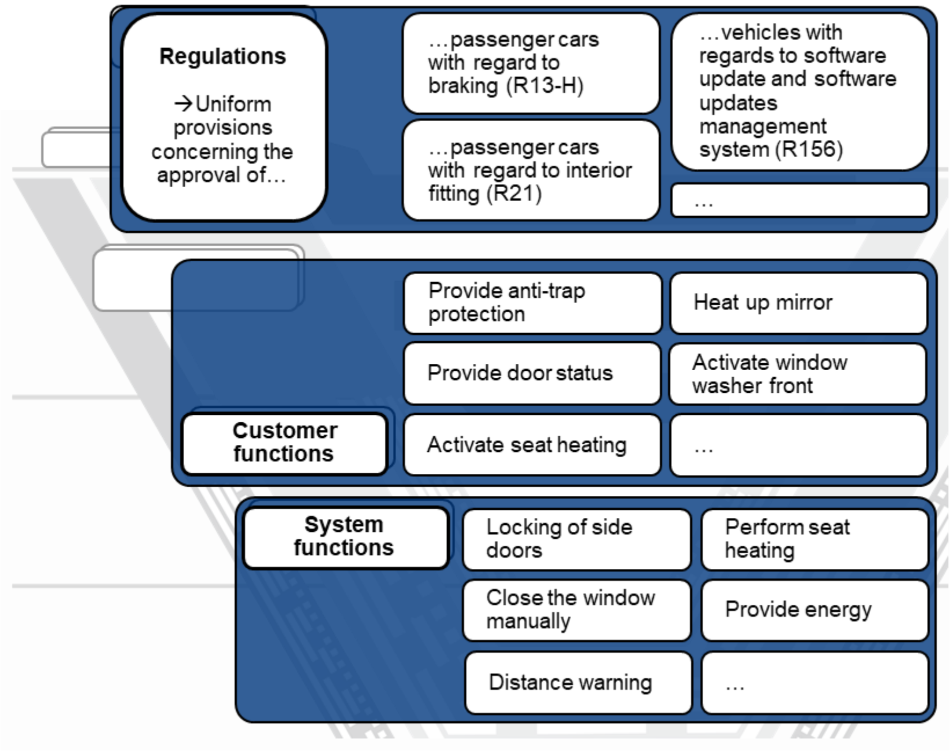
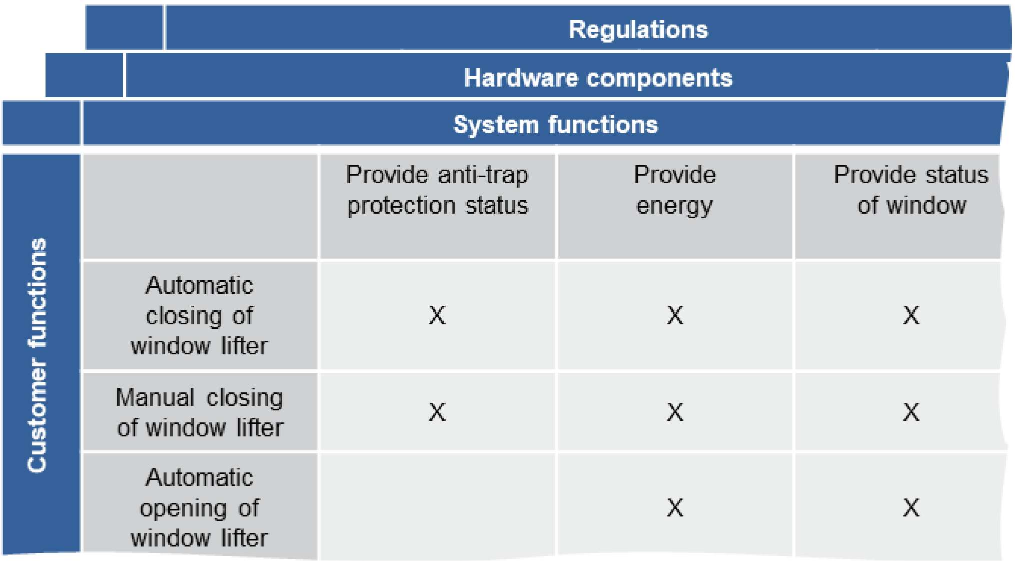

Regulations
→Uniform provisions concerning the approval of…
…passenger cars with regard to braking (R13-H)
…passenger cars with regard to interior fitting (R21)
…vehicles with regards to software update and software updates management system (R156)
…
Customer functions
Provide anti-trap protection
Heat up mirror
Provide door status
Activate window washer front
Activate seat heating
…
System functions
Locking of side doors
Perform seat heating
Close the window manually
Provide energy
Distance warning
…

Figure 11. Provided customer and system functions of a vehicle.

Elicit information: The relations between the trace class artifacts are missing. Therefore, all relations must be elicited in workshops with the experts of the sub-system window lifter using design structure matrixes. As shown in Figure 12, the relations between customer functions, system functions, hardware components, and regulations are elicited in separate matrixes. The design structure matrixes are prefilled with the available information from the provided excel sheets. Exemplary relations are:

- The customer function "Automatic closing of window lifter" is executed by the system functions "Provide anti-trap protection" and "Provide status of window" among other system functions.
- The system function "Provide status of window" is implemented on the ECU "Door module".
- The R 21 demands the system function "Anti-trap protection window lifter".



Regulations

Hardware components

System functions

Customer functions

	Provide anti-trap protection status	Provide energy	Provide status of window
Automatic closing of window lifter	X	X	X
Manual closing of window lifter	X	X	X
Automatic opening of window lifter		X	X

Figure 12. Design Structure Matrix for dependencies between artifact types.

Ensure information quality: To ensure information quality, the systems engineers and certification experts are asked to review the information after the workshop to ensure completeness and correctness.

Step 3: Modeling of effect chains

Model SysML Profile: Before starting to model the effect chains, stereotypes are defined in the SysML profile. Stereotyped extensions of the metaclass <<class>> are <<customer function>>, <<system function>>, <<software component>>, <<hardware component>>, <<regulation>>, and <<certification requirement>>.

Model decomposition: At the beginning of the modeling of the window lifter, the package structure is defined (see Figure 13). Then, the overall model <<vehicle>> is separated into components, sub-systems, and regulations. The functions are represented in detail for each sub-system of the vehicle series. In addition, every system has customer function and system function packages. These packages include additional relevant information about connectors, ports, and other relations.

Model structure: behavior and dependencies: The effect-chain diagram is an ibd where the dependencies between the allocated system functions of a customer function are modeled. Figure 14 shows the effect chain of the customer function "Automatic closing of the window lifter" by the processing system function "Perform automatic closing of window" and results in the information called "Window status", which is passed to the system function "Provide status of window". Represented by the two diagrams req [Certification Requirements] and bdd [Component], relations to certification requirements, regulations, and components are visualized. The relations between system functions and customer functions, as well as between components and regulations and system functions, are modeled via <<allocate>> or <<deriveReqt>> relations. For example, the system function "Provide anti-trap protection status" is allocated to the certification requirement

"Provide anti-trap protection", which is a derived requirement from the regulation "UN ECE R21".

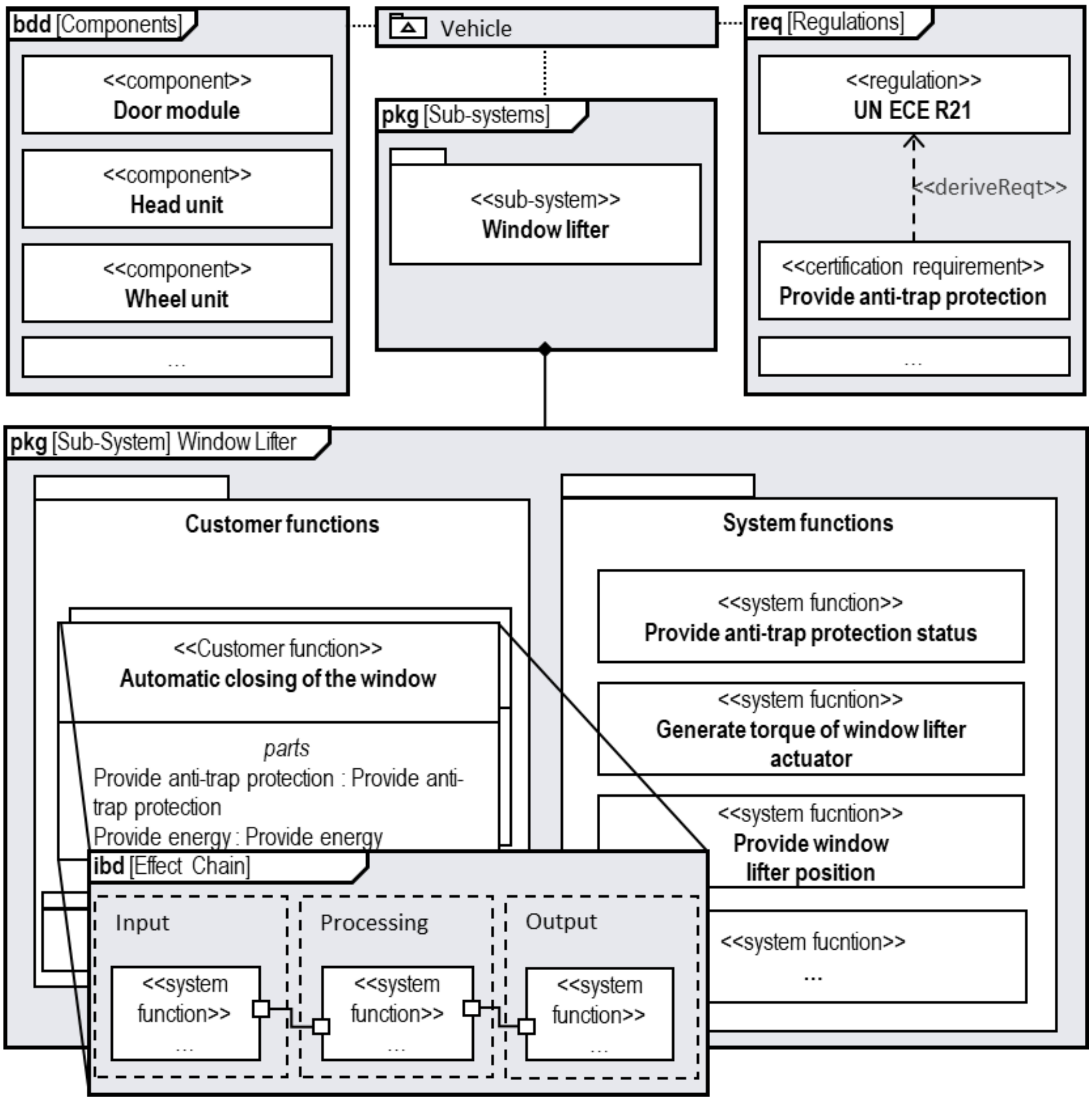


Figure 13. System structure of window lifter.

Step 4: Analysis of effect-chain model

The resulting model can be analyzed to identify regulations, customer functions, and system functions, which are affected by a software change on an ECU of the vehicle series. Additionally, other technical changes can be analyzed. This analysis can be automated using structured expressions or database queries, which analyze the exported SysML information.

6.2. Evaluation of Success Criteria

After the successful demonstration of the application in an industrial case example, the evaluation of the success criteria and premises is conducted. An in-depth evaluation is ensured by the feedback of the interview experts as well as the application experience with the interdisciplinary engineers from the mentioned projects in the automotive domain. Finally, an overview of the evaluation result is presented in Table 2.

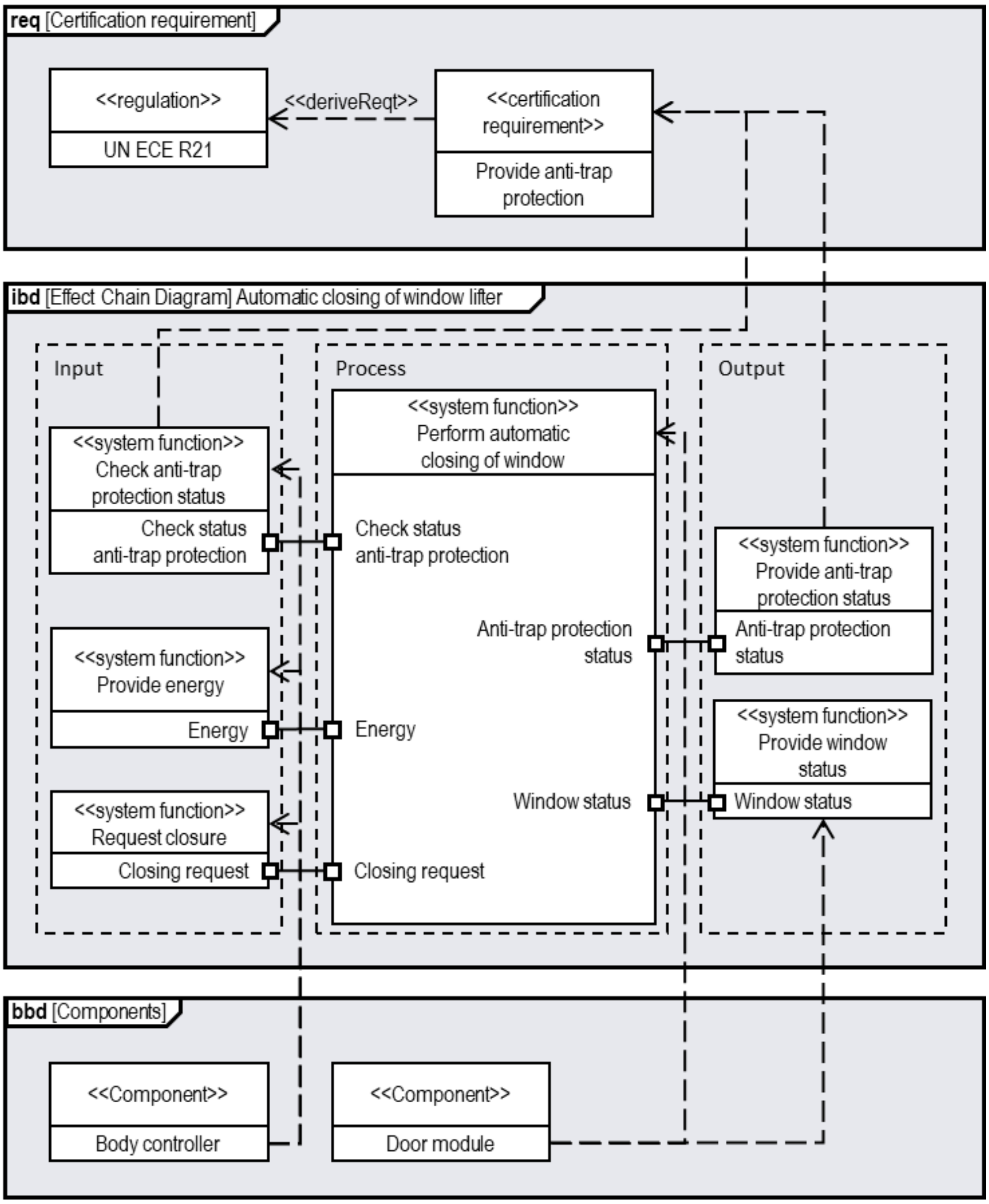

req [Certification requirement]
<<regulation>>
UN ECE R21
<<deriveReqt>>
<<certification requirement>>
Provide anti-trap protection
ibd [Effect Chain Diagram] Automatic closing of window lifter
Input
Process
Output
<<system function>>
Check anti-trap protection status
Check status anti-trap protection
<<system function>>
Provide energy
Energy
<<system function>>
Request closure
Closing request
<<system function>>
Perform automatic closing of window
Check status anti-trap protection
Anti-trap protection status
Energy
Window status
Closing request
<<system function>>
Provide anti-trap protection status
Anti-trap protection status
<<system function>>
Provide window status
Window status
bbd [Components]
<<Component>>
Body controller
<<Component>>
Door module

Figure 14. Effect-chain diagram of "Automatic closing of window lifter".

Table 2. Degree of fulfillment of success criteria and premises.

Input	Fulfillment	Application	Fulfillment
SC-1	•	SC-7	•
SC-2	•	SC-8	◑
SC-3	•	SC-9	◑
SC-4	•	SC-10	•
Output		**Methodology**	
SC-5	•	SC-11	•
SC-6	•	SC-12	•

Key: SC = Success Criteria; • fulfilled; ◑partly fulfilled.

6.3. Evaluation of Input Success Criteria

The methodology must be able to process interdisciplinary artifacts as input for the effect-chain model. With the methodology, typical engineering artifacts (requirements, functions, logical elements, physical elements, verification tests) can be processed. As a tool, the RFLPV handouts can be used to support the modeling experts. Therefore, the methodology can integrate different artifacts as input for the effect-chain model (SC-1). Due to the usage of SysML as a modeling language, the number of processable artifacts is not limited. Compared to other modeling languages, SysML, as a semiformal modeling language, can be adapted to the incoming artifact classes, which supports handling many artifacts. Depending on the number of artifacts, it is suggested to use different views to ensure transparency and clarity (SC-2). The fulfillment of the SC-3 depends on the selection of the modeling tool. Each modeling tool implies a set of tool interfaces and data exchange formats. Typically, SysML models can be exchanged as XMI (Exchange Metadata Interchange) or in mof (Meta Object Facility) formats (SC-3). Information quality must be ensured to model a consistent effect-chain model. Therefore, the information quality is reviewed in step 2 by information quality criteria such as faultlessness or accessibility (SC-4).

6.4. Evaluation of Output Success Criteria

Depending on the modeled elements, effects can be derived. The effects can be evaluated based on expert knowledge or using semi-automized or automized algorithms (SC-5). The accuracy depends on the level of granularity, which is defined in step 1. If the accuracy does not fit the need, the model can be detailed to increase the depth (SC-6).

6.5. Evaluation of Application Success Criteria

In the goal definition (step 1), the regulation analysis depends on the analyzed system. Based on this, the TIM is uniquely defined to ensure compliance with the identified regulations. Therefore, the application of the methodology is independent of the regulation and adapts to the TIN (SC-7). The application effort depends on the number of different artifacts and the required granularity. Therefore, the application effort has to be evaluated in each case example. Additionally, the authors suggest possibilities to automatize the modeling to reduce the application effort as well as put the focus on frontloading within the application of the method (SC-8). The collaborative application depends on the investment in modeling licenses. Using the Cameo Systems Modeler, a collaborative application is possible. D'assault Systèmes provides an additional tool, called Teamwork Cloud, to enable the collaborative usage of the software tool (SC-9). Different tools can support the engineers in applying the methodology and are not limited to the stated examples. One stated example from above is the role model of the MBSE application, which supports engineers in defining participating roles and their responsibilities in using the effect-chain model (step 1). Another one is the RFLPV handouts, which support engineers in the identification of necessary information (step 2). Besides using RFLPV handouts, the quality criteria can be applied to ensure information quality (step 2). The MBSE control questions can be used as an additional tool to ensure the completeness of each step according to the engineering methodology (step 1–step 3). Providing the various supporting elements makes it possible to apply the methodology even without prior knowledge regarding certification-relevant effect-chain modeling (SC-10).

6.6. Evaluation of Methodology Success Criteria

With the methodology, interdisciplinary effects can be represented within typical SysML diagrams. Depending on the TIM, different views and diagrams can be used to model interdisciplinary trace artifact classes and their relations. From mechanical, electrical, and software perspectives, effects between elements are modeled in the diagrams, for example, energy, material, or information flows in an internal block diagram (ibd) (SC-11). For modeling the system's structure in SysML, a block definition diagram, internal block diagram, package diagram, and parametric diagram can be used. For modeling the behavior of systems in SysML, an activity diagram, sequence diagram, state machine diagram, and use case diagram can be used (SC-12).

6.7. Evaluation of Premises

For the application of SysML, different software tools exist. Software tools such as D'assault Systémes Cameo Systems Modeler, Sparx Systems Enterprise Architect, and IBM Rhapsody differ in the provided functionalities and license terms (P-1). To ensure availability and information quality, the authors suggest modeling reoccurring trace artifact classes, such as RFLPV (requirements, functions, logical elements, physical elements, verification tests), if possible. These are typical engineering artifacts and are part of practically relevant engineering methodologies. Therefore, it can be assumed that the information is available explicitly in data formats or implicitly as expert knowledge (P-2). Existing modeling tools (see P-1) are compatible with other engineering tools using standard interfaces such as XMI (P-3).

7. Discussion

The findings of the systematic literature study and the expert interviews were used to identify the limitations of existing approaches. The limitations of existing approaches include applicability for specific regulations, missing tailoring and adaptability, and the lack of supporting tools (see Figure 3). From a practical-oriented point of view, the experts interviewed added the need for frontloading since the employees' experience in model-based systems engineering varies. Furthermore, the experts emphasize that the effort to apply a methodology has to be acceptable in industry. Based on these findings, the new MECA methodology was developed to meet the demands for certification-compliant modeling.

Compared to existing approaches, the MECA methodology is a generic approach that focuses on the early definition of a certification-compliant goal for effect-chain modeling instead of focusing on a specific regulation. The focus on the first step of the methodology supports the frontloading and leads to tailored activities of the following steps, resulting in a need-based set of artifacts and relations. Instead of modeling the whole system with existing approaches, engineers are enabled to derive a need-based TIM, focusing on the certification-compliant effect chains. The approach can be applied to existing software tools and is not limited to a specific tool vendor. In addition, engineers are provided with new tools for the modeling of certification-compliant effect chains such as the RFLPV handouts, control questions, and glossaries. This exceeds the provision of supportive tools compared to existing approaches. Those tools were derived on the basis of the experience of the practical application in industrial projects. An example is the glossary, which was derived from the experience that the earlier a consistent understanding of the modeling context is created, the lower the number of inefficient iterations between the corresponding persons is.

The application of the MECA methodology is demonstrated in the case example, which is a detailed description of the sub-system window lifter of an automotive series. Besides the window lifter, the modeling of effect chains for more than 100 different subsystems of the automotive series underlines the applicability and scalability to systems of varying complexity, such as sensors, actuators, control units, and electric motors. Those sub-systems are also part of systems in other domains such as airplanes. Nevertheless, applicability in other domains needs to be proven in future projects, for example, for robots in a highly regulated medical environment [86,87].

The evaluation underlines the fulfillment of ten of the derived success criteria. This indicates that the limitations of existing approaches have been reduced. Implications for further research exist in the quantitative proof of the application effort and the proof for collaborative modeling in different projects. In addition, there is the possibility to describe individual aspects of the methodology in more detail, for example, the application of information quality criteria and metrics as well as the in-depth description of the connectivity of information artifacts.

8. Conclusions

In the paper, a methodology for the certification-compliant modeling of effect chains is developed. Based on a systematic literature study, success criteria and premises are derived, evaluated, and completed through three interviews with industrial modeling experts. According to the success criteria and premises, the methodology is developed, including methods, models, and tools for the engineers. The applicability is evaluated in an automotive case example. The demands to be fulfilled by a methodology for modeling certification-compliant effect chains in practice are represented by 14 success criteria (RQ1). The evaluation based on the success criteria indicates that the MECA methodology fulfills the demanded needs by combining methods, models, and tools (RQ2). Examples of the combination are the TIM approach, the application of modeling languages, mapping to the V-Model, and RFLPV handouts. Other tools can be included, for example, the main feature list for categorizing requirements [88]. The methodology is tailored to the regulatory demands by defining the TIN and limiting the modeling effort by reducing the modeling effort to modeling necessary model elements and relations according to the needs (RQ3). The methodology offers possibilities for transferring to other domains, such as health industries or aerospace.

Further potential is given by including other existing product data and lifecycle management tools in the underlying toolchain of the MECA methodology [6].

Additionally, artificial intelligence approaches can automatize the identification of relations between system artifacts and reduce the modeling effort [80]. In addition, considering early system artifacts in the effect chain can help to expand traceability and explore the certification-compliant designs in early phases with an engineering approach [89]. Moreover, the certification requirements derived from the given regulations can be integrated into the requirements-driven SE approach to increase the engineered resilience of systems [90]. Additionally, time and costs can be saved by developing a product and a corresponding production system similarly [64]. All in all, the MECA methodology supports engineers in fulfilling regulations by modeling effect chains, which serves as a foundation for increased modeling capabilities and complying with other regulations.

Author Contributions: Conceptualization, I.G. and D.W.; methodology, I.G. and D.W.; validation, D.W., A.-S.K. and T.M.; formal analysis, D.W., A.-S.K. and T.S.; investigation, I.G., T.S., T.M. and D.W.; resources, I.G. and T.S.; writing—original draft preparation, D.W. and A.-S.K.; writing—review and editing, I.G., T.S. and T.M.; visualization, D.W. and A.-S.K.; supervision, I.G.; project administration, D.W. and T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all interviewees involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy reasons.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. International Council on Systems Engineering. *Systems Engineering Vision 2035, Engineering Solutions for a better World*; T INCOSE Central Office: San Diego, USA, 2021.
2. Gräßler, I. Umsetzungsorientierte Synthese mechatronischer Referenzmodelle: Implementation-oriented synthesis of mechatronic reference models. In *Konferenzband der VDI Mechatronik*; Fachtagung Mechatronik: Dortmund, Germany, 12–13 March 2015; pp. 167–172.
3. Gräßler, I.; Wiechel, D.; Koch, A.-S.; Preuß, D.; Oleff, C. Model-based effect-chain analysis for complex systems. In Proceedings of the 17th International Design Conference, Cavtat, Croatia, 23–26 May 2022; Cambridge University Press: Cambridge, UK, 2022.

4. Rempel, P.; Mäder, P.; Kuschke, T.; Cleland-Huang, J. Mind the gap: Assessing the conformance of software traceability to relevant guidelines. In *Proceedings of the ICSE '14: 36th International Conference on Software Engineering, Hyderabad, India, 31 May–7 June 2014*; Jalote, P., Briand, L., van der Hoek, A., Eds.; ACM: New York, NY, USA, 2014; pp. 943–954. ISBN 9781450327565.
5. Regan, G.; Biro, M.; Flood, D.; McCaffery, F. Assessing traceability-practical experiences and lessons learned. *J. Softw. Evol. Proc.* **2015**, *27*, 591–601. [CrossRef]
6. Štorga, M.; Marjanović, D.; Savšek, T. Reference model for traceability records implementation in engineering design environment. In Proceedings of the ICED 11–18th International Conference on Engineering Design: Impacting Society Through Engineering Design, Lyngby/Copenhagen, Denmark, 15–19 August 2011; pp. 173–182.
7. *610.12-1990 IEEE*; Standard Glossary of Software Engineering Terminology. IEEE/Institute of Electrical and Electronics Engineers Incorporated: Piscataway, NJ, USA, 1983; ISBN 978-0-7381-0391-4.
8. VDA QMC Working Group 13/Automotive SIG. In Proceedings of the Automotive SPICE: Process Reference Model Process Assessment Model, 656th ed. (Automotive SPICE Version 3.1), Berlin, Germany, 2017. Available online: https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf (accessed on 2 February 2023).
9. Nair, S.; de La Vara, J.L.; Sen, S. A review of traceability research at the requirements engineering conferencere@21. In *Proceedings of the 2013 IEEE 21st International Requirements Engineering Conference (RE), Rio de Janeiro, RJ, Brazil, 15–19 July 2013*; IEEE: Piscataway, NJ, USA, 2013; pp. 222–229. ISBN 978-1-4673-5765-4.
10. Gräβler, I.; Pöhler, A. Produktentstehung im Zeitalter von Industrie 4.0. In *Handbuch Gestaltung Digitaler und Vernetzter Arbeitswelten*; Maier, G.W., Engels, G., Steffen, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 383–403. ISBN 978-3-662-52898-3.
11. Dean, J.; Henderson, C.; Gardner, J. Model-Based Systems Engineering as an Enabler for Regulatory Design Compliance. *INCOSE Int. Symp.* **2012**, *22*, 2266–2278. [CrossRef]
12. Laufenberg, L. *Methodik zur Integrierten Projektgestaltung für die Situative Umsetzung des Simultaneous Engineering*; Shaker: Aachen, Germany, 1996. (In Germany)
13. Gräβler, I. *Informations-und zeitbasiertes Controlling einer Integrierten Konstruktion und Arbeitsplanung, Als Ms. gedr*; Shaker: Aachen, Germany, 2000; ISBN 3-8265-7011-1. (In Germany)
14. Ulrich, H. Anwendungsorientierte Wissenschaft. *Die Unternehm.* **1982**, *36*, 1–10. (In Germany)
15. Blessing, L.T.M.; Chakrabarti, A. *DRM, a Design Research Methodology, 1. Auflage*; Springer: London, UK, 2009; ISBN 978-1-84882-587-1.
16. Andrianarison, E.; Piques, J.-D. SysML for embedded automotive Systems: A practical approach. In Proceedings of the ERTS2, Embedded Real Time Software & Systems, Toulouse, France, May 2010.
17. Amalfitano, D.; de Simone, V.; Maietta, R.R.; Scala, S.; Fasolino, A.R. Using tool integration for improving traceability management testing processes: An automotive industrial experience. *J. Softw. Evol. Proc.* **2019**, *31*, e2171. [CrossRef]
18. *AS 9100: (R) Quality Management Systems-Requirements for Aviation, Space, Rev. D.*; SAE International: Warrendale, PA, USA, 2016.
19. Barbosa, P.; Leite, F.; Santos, D.; Figueiredo, A.; Galdino, K. Introducing Traceability Information Models in Connected Health Projects. In *Proceedings of the 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), Karlstad, Sweden, 18–21 June 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 18–23. ISBN 978-1-5386-6060-7.
20. Brandejsky, T. Problems of En 50 128: 2011 Railway Standard. In *Scientific Student Conference Interoperability of Railway Transport-(iricon 2016)*; Jirova, J., Reznickova, J., Eds.; Czech Technical Univ Prague: Prague, Czechia, 2016; pp. 1–3. ISBN 978-80-01-06022-3.
21. *ISO/IEC/IEEE 24765 2010(E)*; ISO/IEC/IEEE International Standard-Systems and Software Engineering–Vocabulary. IEEE: Piscataway, NJ, USA, 2010; pp. 1–418. [CrossRef]
22. Holtmann, J.; Steghofer, J.-P.; Rath, M.; Schmelter, D. Cutting through the Jungle: Disambiguating Model-based Traceability Terminology. In *Proceedings of the 2020 IEEE 28th International Requirements Engineering Conference (RE), Zurich, Switzerland, 31 August–4 September 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 8–19. ISBN 978-1-7281-7438-9.
23. Gotel, O.; Finkelstein, C.W. An analysis of the requirements traceability problem. In *Proceedings of the IEEE International Conference on Requirements Engineering, Colorado Springs, CO, USA, 18–22 April 1994*; IEEE Comput. Soc. Press: Los Alamitos, CA, USA, 1994; pp. 94–101, ISBN 0-8186-5480-5.
24. Hamilton, V.L.; Beeby, M.L. Issues of traceability in integrating tools. In Proceedings of the IEE Colloquium on Tools and Techniques for Maintaining Traceability During Design, London, UK, 2 December 1991; IET: Hong Kong, China.
25. Storga, M. Traceability in product development. In Proceedings of the Design 2004: The 8th International Design Conference, Dubrovnik, Croatia, 18–21 May 2004; pp. 911–918.
26. Rempel, P.; Mader, P. A quality model for the systematic assessment of requirements traceability. In *Proceedings of the 2015 IEEE 23rd International Requirements Engineering Conference (RE), Ottawa, ON, Canada, 24–28 August 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 176–185. ISBN 978-1-4673-6905-3.
27. Gotel, O.; Cleland-Huang, J.; Hayes, J.H.; Zisman, A.; Egyed, A.; Grünbacher, P.; Dekhtyar, A.; Antoniol, G.; Maletic, J.; Mäder, P. Traceability Fundamentals. In *Software and Systems Traceability*; Cleland-Huang, J., Gotel, O., Zisman, A., Eds.; Springer: London, UK, 2012; pp. 3–22. ISBN 978-1-4471-2238-8.
28. Lindvall, M.; Sandahl, K. Practical Implications of Traceability. *Softw. Pract. Exper.* **1996**, *26*, 1161–1180. [CrossRef]
29. Mader, P.; Gotel, O.; Philippow, I. Getting back to basics: Promoting the use of a traceability information model in practice. In *2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering*; IEEE: Piscataway, NJ, USA, 2009.

30. Wohlrab, R.; Steghofer, J.-P.; Knauss, E.; Maro, S.; Anjorin, A. Collaborative Traceability Management: Challenges and Opportunities. In *Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference (RE), Beijing, China, 12–16 September 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 216–225. ISBN 978-1-5090-4121-3.
31. Mader, P.; Gotel, O.; Philippow, I. Motivation Matters in the Traceability Trenches. In *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference (RE), Atlanta, GA, USA, 31 August–4 September 2009*; IEEE: Piscataway, NJ, USA, 2009; pp. 143–148, ISBN 978-0-7695-3761-0.
32. Ramesh, B.; Jarke, M. Toward reference models for requirements traceability. *IIEEE Trans. Softw. Eng.* **2001**, *27*, 58–93. [CrossRef]
33. Aizenbud-Reshef, N.; Nolan, B.T.; Rubin, J.; Shaham-Gafni, Y. Model traceability. *IBM Syst. J.* **2006**, *45*, 515–526. [CrossRef]
34. Delligatti, L. *SysML Distilled: A Brief Guide to the Systems Modeling Language*; Addison-Wesley: Upper Saddle River, NJ, USA; New York, NY, USA, 2014; ISBN 978-0-321-92786-6.
35. International Council on Systems Engineering. *INCOSE Systems Engineering Vision 2020*; INCOSE Central Office: San Diego, CA, USA, 2007.
36. Object Management Group. OMG: Unified Modeling Language Infrastructure Specification, Version 2.0. Available online: https://www.omg.org/spec/UML/2.0/Superstructure/PDF (accessed on 12 October 2020).
37. Object Management Group. OMG Systems Modeling Language (OMG SysML™). Available online: https://sysml.org/.res/docs/specs/OMGSysML-v1.6-19-11-01.pdf (accessed on 30 January 2023).
38. Friedenthal, S.; Moore, A.; Steiner, R. *A Practical Guide to SysML: The Systems Modeling Language*, 3rd ed.; Elsevier: Amsterdam, The Netherlands; Morgan Kaufmann: Waltham, MA, USA, 2015; ISBN 9780128002025.
39. Weilkiens, T. *Systems Engineering mit SysML/UML: Modellierung, Analyse, Design*, 3rd ed.; dpunkt: Heidelberg, Germany, 2014; ISBN 9783864900914.
40. Bajaj, M.; Backhaus, J.; Walden, T.; Waikar, M.; Zwemer, D.; Schreiber, C.; Issa, G.; Martin, L. Graph-Based Digital Blueprint for Model Based Engineering of Complex Systems. *INCOSE Int. Symp.* **2017**, *27*, 151–169. [CrossRef]
41. Holt, J. *Systems Engineering Demystified: A Practitioner's Handbook for Developing Complex Systems Using a Model-Based Approach*; Packt Publishing: Birmingham, UK; Mumbai, India, 2021; ISBN 978-1-83898-580-6.
42. Chami, M.; Bruel, J.-M. *A Survey on MBSE Adoption Challenges*; Wiley Interscience Publications: Hoboken, NJ, USA, 2018.
43. Dick, J.; Hull, E.; Jackson, K. DOORS: A Tool to Manage Requirements. In *Requirements Engineering*; Dick, J., Hull, E., Jackson, K., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 187–206. ISBN 978-3-319-61072-6.
44. Lavazza, L.; Valetto, G. Enhancing requirements and change management through process modelling and measurement. In Proceedings of the ICRE2000: IEEE International Conference on Requirements Engineering, Schaumburg, IL, USA, 19–23 June 2000; pp. 106–115.
45. Pinheiro, F.; Goguen, J.A. An object-oriented tool for tracing requirements. *IEEE Softw.* **1996**, *13*, 52–64. [CrossRef]
46. Bode, S.; Lehnert, S.; Riebisch, M. Comprehensive model integration for dependency identification with EMFTrace. In Proceedings of the MDSM2011 and SQM 2011, Ilmenau, Germany, March 2011; pp. 17–20.
47. Hegedüs, Á.; Horváth, Á.; Ráth, I.; Starr, R.R.; Varró, D. Query-driven soft traceability links for models. *Softw. Syst. Model.* **2016**, *15*, 733–756. [CrossRef]
48. Smith, T.J. READS: A requirements engineering tool. In *Proceedings of the [1993] IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, 4–6 January 1993*; IEEE Computer Society Press: Los Alamitos, CA, USA, 1993; pp. 94–97, ISBN 0-8186-3120-1.
49. Weilkiens, T.; Berres, A.; Endler, D.; Haarer, A.; Lalitsch-Schneider, C.; Krammer, M.; Martin, H. System Safety in SysML. In *Tag des Systems Engineering: Ulm, 11–13 November 2015*; Schulze, S.-O., Muggeo, C., Eds.; Hanser: München, Germany, 2016; ISBN 9783446447288.
50. David, P.; Shawky, M. Supporting ISO 26262 with SysML, Benefits and Limits. In Proceedings of the ESREL, Rhodes, Greece, September 2010; p. 8.
51. Valdivia Dabringer, M.L.; Dybov, A.; Fresemann, C.; Stark, R. Towards Integrated Safety Analysis as Part of Traceable Model-Based Systems Engineering. *Proc. Des. Soc.* **2022**, *2*, 2005–2014. [CrossRef]
52. Hillenbrand, M. *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen*; Zugl.: Karlsruhe, Germany; KIT Scientific Publishing: Karlsruhe, Germany, 2012; ISBN 978-3-86644-803-2.
53. Bleu-Laine, M.-H.; Bendarkar, M.V.; Xie, J.; Briceno, S.I.; Mavris, D.N. A Model-Based System Engineering Approach to Normal Category Airplane Airworthiness Certification. In *Proceedings of the AIAA Aviation 2019 Forum, Dallas, TX, USA, 17–21 June 2019*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2019; ISBN 978-1-62410-589-0.
54. Zhang, J.; Zhang, X.; Li, H. Requirements Engineering Method Applied in the Civil Aircraft Airframe Segment Development. In *Proceedings of the 2020 6th International Conference on Mechanical Engineering and Automation Science (ICMEAS), Moscow, Russia, 29–31 October 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 281–286. ISBN 978-1-7281-9272-7.
55. Glinski, S.; Fazal, B.; Harrison, E.D.; Bendarkar, M.V.; Fields, T.; Garcia, E.; Mavris, D.N. An MBSE Framework to Identify Regulatory Gaps for Electrified Transport Aircraft. In *Proceedings of the 2022 IEEE Transportation Electrification Conference & Expo (ITEC), 2022 IEEE/AIAA Transportation Electrification Conference and Electric Aircraft Technologies Symposium (ITEC+EATS), Anaheim, CA, USA, 15–17 June 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 772–777, ISBN 978-1-6654-0560-7.

56. Sannier, N.; Baudry, B.; Nguyen, T. Formalizing standards and regulations variability in longlife projects. A challenge for Model-driven engineering. In *Proceedings of the 2011 Model-Driven Requirements Engineering Workshop (MoDRE), Trento, Italy, 29 August 2011*; IEEE: Piscataway, NJ, USA, 2011; pp. 64–73, ISBN 978-1-4577-0957-9.
57. Hamraz, B. *Engineering Change Modelling Using a Function-Behaviour-Structure Scheme*; Apollo-University of Cambridge Repository: Cambridge, MA, USA, 2013.
58. Walden, D.D.; Roedler, G.J.; Forsberg, K.; Hamelin, R.D.; Shortell, T.M. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed.; Wiley: Hoboken, NJ, USA, 2015; ISBN 9781118999400.
59. Gräβler, I.; Wiechel, D.; Oleff, C. Extended RFLP for complex technical systems. In Proceedings of the 8th IEEE International Symposium on Systems Engineering, ISSE, Vienna, Austria, 24–26 October 2021; IEEE: Piscataway, NJ, USA, 2022.
60. Gräβler, I.; Wiechel, D.; Pottebaum, J. Role model of model-based systems engineering application. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1097*, 12003. [CrossRef]
61. VDI/VDE. *Entwicklung Mechatronischer und Cyber-Physischer Systeme*; Beuth Verlag GmbH: Düsseldorf, Germany, 2021. (In Germany)
62. Graessler, I.; Hentze, J. The new V-Model of VDI 2206 and its validation. *At-Automatisierungstechnik* **2020**, *68*, 312–324. [CrossRef]
63. Gräβler, I.; Hentze, J.; Bruckmann, T. V-Models for Interdisciplinary Systems Engineering. In Proceedings of the DESIGN 2018 15th International Design Conference, DESIGN Conference, Dubrovnik, Croatian, 21–24 May 2018; Design Society: Singapore; pp. 747–756.
64. Gräβler, I.; Hentze, J.; Yang, X. Eleven Potentials for Mechatronic V-Model. In *Proceedings of the International Conference Production Engineering and Management*; Lemgo, Germany, 29–30 September 2016, Villmer, F.-J., Padoanao, E., Eds.; 2016; pp. 257–268, ISBN 978-3-946856-00-9.
65. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*; MIT Press: Cambridge, MA, USA, 2012; ISBN 9780262017527.
66. Legard, R.; Keegan, J.; Ward, K. In-depth interviews. In *Qualitative Research Practice: A Guide for Social Science Students and Researchers, Repr*; Ritchie, J., Ed.; SAGE: Los Angeles, CA, USA, 2011; pp. 138–169. ISBN 9780761971108.
67. Powell, R.A.; Single, H.M. Focus groups. *Int. J. Qual. Health Care* **1996**, *8*, 499–504. [CrossRef] [PubMed]
68. Stvilia, B.; Gasser, L.; Twidale, M.B.; Smith, L.C. A framework for information quality assessment. *J. Am. Soc. Inf. Sci.* **2007**, *58*, 1720–1733. [CrossRef]
69. Krcmar, H. *Informationsmanagement, 6., überarb. Aufl.*; Springer Gabler: Wiesbaden, Germany, 2015; ISBN 978-3-662-45863-1. (In Germany)
70. Rohweder, J.P.; Kasten, G.; Malzahn, D.; Piro, A.; Schmid, J. Informationsqualität—Definitionen, Dimensionen und Begriffe. In *Daten-und Informationsqualität*; Hildebrand, K., Gebauer, M., Hinrichs, H., Mielke, M., Eds.; Vieweg+Teubner: Wiesbaden, Germany, 2008; pp. 25–45, ISBN 978-3-8348-0321-4. (In Germany)
71. Wang, R.Y.; Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manag. Inf. Syst.* **1996**, *12*, 5–33. [CrossRef]
72. Gräβler, I. A new V-Model for interdisciplinary product engineering. In *Engineering for a Changing World, 59th IWK*; Ilmenau Scientific Colloquium: Ilmenau, Germany, 11–15 September 2017.
73. Kleiner, S.; Kramer, C. Model Based Design with Systems Engineering Based on RFLP Using V6. In *Smart Product Engineering, Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, 11–13 March 2013*; Abramovici, M., Stark, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 93–102, ISBN 9783642308161.
74. Haberfellner, R.; de Weck, O.L.; Fricke, E. *Systems Engineering: Fundamentals and Applications*; Birkhäuser: Basel, Switzerland, 2019; ISBN 978-3-030-13430-3.
75. Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H. *Engineering Design*; Springer: London, UK, 2007; ISBN 978-1-84628-318-5.
76. Stachowiak, H. *Allgemeine Modelltheorie*; Springer: Wien, Germany, 1973; ISBN 3-211-81106-0.
77. Gräβler, I.; Wiechel, D.; Thiele, H. Fortschrittskontrolle der Modellierung mechatronischer Produkte: Controlling of the Modeling of Mechatronic Products. In *Fachtagung VDI Mechatronik 2022*; Bertram, T., Corves, B., Janschek, K., Rinderknecht, S., Eds.; Technische Universität Darmstadt: Darmstadt, Germany, 2022; pp. 49–54.
78. Gräβler, I.; Oleff, C.; Scholle, P. Method for Systematic Assessment of Requirement Change Risk in Industrial Practice. *Appl. Sci.* **2020**, *10*, 8697. [CrossRef]
79. Menninger, B.; Wiechel, D.; Rackow, S.; Höpfner, G.; Oleff, C.; Berroth, J.; Gräßler, I.; Jacobs, G. Modeling and analysis of functional variance of complex technical systems. In Proceedings of the 33rd Symposium Design for X, Hamburg, Germany, 22–23 September 2022; The Design Society: Glasgow, UK, 2022; p. 10.
80. Gräβler, I.; Oleff, C.; Hieb, M.; Preuβ, D. Automated Requirement Dependency Analysis for Complex Technical Systems. *Proc. Des. Soc.* **2022**, *2*, 1865–1874. [CrossRef]
81. Gräβler, I.; Oleff, C.; Preuβ, D. Proactive Management of Requirement Changes in the Development of Complex Technical Systems. *Appl. Sci.* **2022**, *12*, 1874. [CrossRef]
82. de-academic.com. Elektrischer Fensterheber. Available online: https://de-academic.com/dic.nsf/dewiki/384632 (accessed on 26 October 2022).

83. Jacobs, G.; Konrad, C.; Berroth, J.; Zerwas, T.; Höpfner, G.; Spütz, K. Function-Oriented Model-Based Product Development. In *Design Methodology for Future Products*; Krause, D., Heyden, E., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 243–263. ISBN 978-3-030-78367-9.
84. United Nations Economic Commission for Europe. UN Regulation No. 156: Uniform Provisions Concerning the Approval of Vehicles with Regards to Software Update and Software Updates Management System. Available online: https://unece.org/sites/default/files/2021-03/R156e.pdf. (accessed on 30 January 2023).
85. United Nations Economic Commission for Europe. UN Regulation No. 21: Uniform Provisions Concerning the Approval of Vehicles with Regards to Their Interior Fittings. Available online: https://unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/r021r2e_1.pdf. (accessed on 30 January 2023).
86. Qi, W.; Ovur, S.E.; Li, Z.; Marzullo, A.; Song, R. Multi-Sensor Guided Hand Gesture Recognition for a Teleoperated Robot Using a Recurrent Neural Network. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6039–6045. [CrossRef]
87. Su, H.; Mariani, A.; Ovur, S.E.; Menciassi, A.; Ferrigno, G.; de Momi, E. Toward Teaching by Demonstration for Robot-Assisted Minimally Invasive Surgery. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 484–494. [CrossRef]
88. Gräβler, I.; Dattner, M.; Bothen, M.; Bothen, M. Main Feature List as core success criteria of organizing Requirements Elicitation. In Proceedings of the R&D Management Conference, Mailand, Italy, 30 June–4 July 2018; pp. 1–16.
89. Specking, E.; Parnell, G.; Pohl, E.; Buchanan, R. Early Design Space Exploration with Model-Based System Engineering and Set-Based Design. *Systems* **2018**, *6*, 45. [CrossRef]
90. Ferris, T.L.; Specking, E.; Jackson, S.; Parnell, G.; Pohl, E. The Fundamental Nature of Resilience of Engineered Systems. *INCOSE Int. Symp.* **2018**, *28*, 1311–1321. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.