INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.191

Model-Based Effect-Chain Analysis for Complex Systems

I. Gräßler, D. Wiechel ✉, A.-S. Koch, D. Preuß and C. Oleff
Heinz Nixdorf Institute, Paderborn University, Germany
✉ dominik.wiechel@hni.uni-paderborn.de

Abstract

Effect chain modeling approaches are applied to model cause-effect relations and analyze affected elements and dependencies. In this paper a systematic literature research is conducted to derive main characteristics and limitations of existing approaches. Then, the Model-based Effect Chain Analysis (MECA) method is introduced. Evaluation proves applicability of the method by means of a case example. This is done in the context of a project with a German automotive company. In the project 66 workshops were conducted to model certification-compliant effect chains in accordance to the UN ECE 156.

Keywords: model-based systems engineering (MBSE), modelling, systems engineering (SE)

1. Introduction

Complexity in the development of cyber-physical systems rises in different areas: (1) the cyber-physical system itself, (2) the company organisation including processes, internal communication and company structure, and (3) the business environment including suppliers, customers and market regulators (Hamraz 2013). Additionally, innovative technologies are leading to an increase in functionalities, for example e-mobility and autonomous driving in modern vehicles (Chamas and Paetzold 2018). This results in additional complexity within technical systems as well as in an increasing number of interactions between system elements of the vehicle, which must be considered in development (see Figure 1). For example, the interaction between different subsystems of the vehicle. One example is the interaction of the window lifter and climate control to the user-commands from devices like a smart key. All systems have their own development artefacts (requirements, functions, logical elements, physical elements) and are surrounded by additional artefacts of the system context like, for example, regulations. Due to the complex interaction, changes to system elements not only have immediate effects, but propagate within the overall system and across system boundaries, leading to increasing costs, changes in quality or even project failure (Eger et al. 2007). The resulting effects can hardly be evaluated without a systematic approach and software support. Impact analysis can be used to analyze existing information and to help developers estimate the impacts of occurring engineering change requests (Gräßler and Wiechel 2021). Effect-chain modeling is one way to build up such an information model. The necessary modeling languages and tools are already available for implementation. However, existing modeling methods only consider specific aspects such as functional-chain modeling (Alt 2012) of the mechatronic products, without including upstream or downstream artefacts in the framework under consideration. This leads to the following research question: *How do existing modeling methods need to be adapted to enable systematic modeling and analysis of effect chains for complex technical systems?*
The paper at hand is structured as follows: in Section 2, the research approach is presented which is designed to answer the research question. Section 3 describes the related work to the stated topic. In

Section 4, the MECA-method is described. The evaluation of the method is described in Section 5. Section 6 summarises the conclusions and provides an outlook.

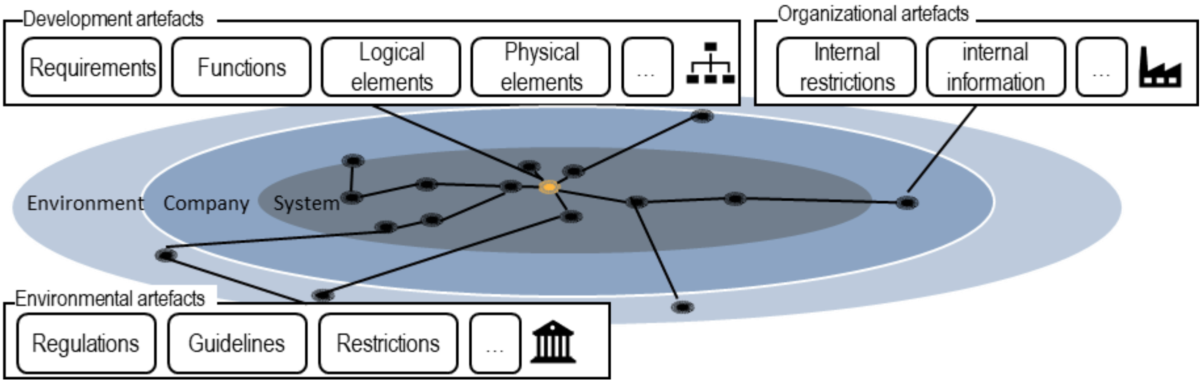

Figure 1. Interconnected artefacts within a development process of complex technical systems

2. Research approach

The research approach consists of four steps (see Figure 2). First, a systematic literature research (Machi and McEvoy 2012) is executed according to keywords such as "MBSE", "effect chain" or "model based" in English and German language. Second, the results of the research are analyzed regarding methods for modeling and analyzing effect chains. Elements are categorized according to contextual focus and clustered regarding their contribution. Main characteristics and limitations of the approaches for modeling and analyzing effect chains are derived. Third, the MECA (Model-based Effect-Chain Analysis) method is developed. Constituents of the method are defined by adapting existing elements towards the application context. To illustrate application of the method and foster understanding, a case example of a German Automotive company regarding certification-compliant modeling is presented. Application and Success Evaluation (Blessing and Chakrabarti 2009) is carried out in the context of a project on effect-chain modeling to fulfil the UN ECE 156 for 30 systems of an automotive series project. Within the project, 36 workshops were conducted with 25 industry experts for data collection. 30 additional workshops were executed for validation of completeness and correctness of effect chains.

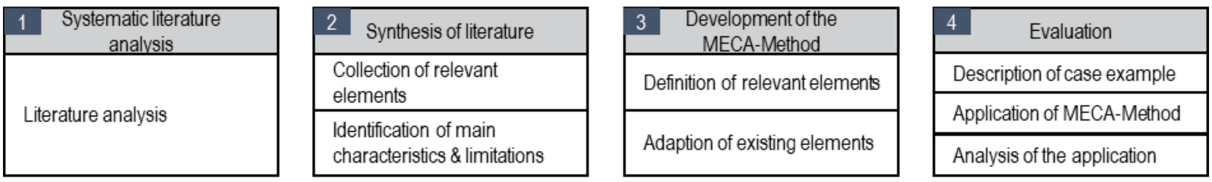

Figure 2. Research approach

3. Related work

Systems are an integrated set of elements that accomplish a defined objective. The complexity of modern systems results from the increased number of system elements and the amount of information and knowledge needed to describe the system (Gräßler 2015). Model-based Systems Engineering is an approach to formalize the development of complex technical systems within a system model. MBSE is defined as "the formalized application of modeling to support systems requirements, design, analysis, verification and validation activities beginning in the conceptual phase and continuing throughout development and later life cycle phases" (International Council on Systems Engineering 2007). A system model is a reliable source of information, which ensures consistency between requirements, design, analysis, and verification (Estefan 2008). In dependency matrices each cell contains numeric or binary information about the dependency between the considered artefact of the row and column (Clarkson et al. 2004). This results in an exponentially growing number of cells to be filled per artefact added. Examples of dependency matrices are the Design Structure Matrix (DSM) and Domain

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

Mapping Matrix (DMM), which are square matrices including components in rows and columns (DSM) or artifacts (DMM) from different domains (e.g. tasks and persons) (Danilovic and Browning 2007). Depending on the complexity of the system, the application effort to manually fill these matrices exceeds a reasonable level (Gräßler et al. 2022). An alternative approach to map artifacts and capture dependencies is a system model. A system model can be defined by using (1) a modeling method, (2) a modeling language and (3) a modeling tool (Delligatti 2014). Different methods describe the procedure to model a complex system, for example OOSEM (Friedenthal *et al.*, 2008) and SYSMOD (Weilkiens 2014). Additionally, different modeling languages can be used to describe the system, for example UML (Object Management Group 2004) and SysML (Object Management Group 2019). Various IT vendors offer tools to execute SysML, for example IBM Rhapsody or Cameo Systems Modeler from Dassault Systems. The application of MBSE helps to reduce discipline-specific thinking and lacks of transparency. One major benefit of executable models is the possibility to create a set of dependencies between different system models, which ensures traceability and can be analyzed. Traceability can be modeled as bidirectional dependency between system elements including design characteristics, design enablers as well as system requirements (Walden et al. 2015). Beside vertical traceability (assignment of development artefacts), the artefacts can be assigned horizontally across disciplines (Hick et al. 2019).

One major problem of MBSE is the high modeling effort to create a consistent, correct and complete system model. Therefore, tailoring and scoping of the modeling effort is key, to ensure a positive ratio of benefits of mapping the elements against modeling effort (Friedenthal *et al.*, 2008). Effect-chain modeling has the intention to systematically identify and map the smallest possible set of necessary elements and dependencies to capture specific cause-effect relationships. Therefore, effect chain modeling can be seen as a another view of a system model, which can be used in addition or even alternatively. Model-based effect chains enable mapping and interpretation of dependencies between a specific set of system elements of an overall system. In the aerospace or railway sectors, tools such as traceability matrices are used to verify safety-related requirements in order to comply with existing standards (Kirova et al. 2008). With a mechanical point of view, Frei (Frei, 2000) defines effect chains on the basis of VDI 2221 (Guideline VDI 2221:2019) as a combination of affected areas of a technical system, which enforce the implementation of the selected physical effects and can be determined on the basis of (partial) functions. Alt describes effect chains as a network of different components focusing on the functional perspective of complex systems. His approach includes a division of the considered elements into input, process and output elements (Alt 2012). Albers and Zingel (Albers and Zingel 2013) describe effect chains with a MBSE-perspective as a certain sequence of functions. Kramer and Münzenberger describe effect chains from an electrical perspective as the data path through the system, where the data transfer and data processing is located (Kramer and Münzenberger 2010).

3.1. Literature review and definition of effect chains

Based on these discipline specific definitions, a definition of effect chains of complex interdisciplinary systems is derived: **Model-based effect chains of interdisciplinary complex systems are cause-effect relationships that are mapped using traceability and system element artefacts.** Based on this understanding, a comprehensive literature review in accordance to Machi & McEvoy (Machi and McEvoy 2012) is conducted. The focus of the literature review is to identify approaches for effect-chain modeling and analysis. Due to the large number of results, a reading scheme is applied to select relevant literature (Blessing and Chakrabarti 2009). Based on the reading scheme and K.O. criteria like "reference to technical systems", 16 approaches are selected. The results are illustrated in Table 1. Different approaches define technical terms and procedures to model and analyze effect chains (Dobrusskin 2016). Alt's approach focuses on the model-based top-down description of technical systems. In his approach, he focuses on functional chains, which are built according to the IPO logic (input, processing, output) (Alt 2012). A similar structure of elements and diagrams is used on other abstraction levels of the system model, which results in a limitation to the functional view. Chamas and Paetzold (Chamas and Paetzold 2018) model effect chains by deriving customer functions from customer requirements. Subsequently, functional, behavioural and structural views are determined, resulting in the required information for effect-chain modeling. This approach limits the beginning of the effect chain by the need to define customer requirements as the starting point. Meyer et al. (Meyer et al. 2011) define an approach that allows a formal specification analysis of timing requirements within an effect chain for software systems. All approaches are analyzed in a synthesis of literature.

Table 1. Literature review for effect-chain modeling

Database	Keywords	Results	Relevant literature
IEEE Explore	("SysML" OR "MBSE" OR "model based") AND ("effect chain" OR "impact chain")	1	(Becker and Mubeen, 2018)
Google Scholar	("modellierung" OR "modelbasiert") AND ("wirkkette") OR "Wirkketten-Modellierung"	328	(Broy et al. 2011; Frei 2000; Kramer and Münzenberger 2010; Feilhauer 2018)
	("MBSE" OR "SysML") AND ("Wirkkette" OR "Wirkketten")	38	(Schmitt 2020; Meyer et al. 2011)
	allintitle: ("effect chain" OR "impact chain")	90	(Lee et al. 2018; Cao et al. 2006; Dobrusskin 2016)
	("MBSE" OR "SysML") AND ("effect chain" OR "impact chain")	51	(Schwede et al. 2019; Albers and Zingel 2013; Kaiser et al. 2010; Song et al. 2010 - 2010; Zhao 2017)
	"impact analysis" AND "interdisciplinary" AND "effect chain" AND "MBSE"	3	(Chamas and Paetzold 2018)
Design Society	+effect chain +(>SysML<MBSE<model based)	132	(Hackl and Krause 2017)
Total		**725**	**16**

3.2. Synthesis of literature

Based on the relevant elements of these approaches, three main characteristics are identified:

- **Deliberate system model:** One major characteristic of an effect chain is the deliberated view of the system model. When modeling the effect chain, the engineer focuses on a subset of system elements and dependencies, for example functions (Chamas and Paetzold 2018; Alt 2012) and their realization by components (Alt 2012).
- **Starting-point and ending-point:** An effect chain has a starting point and an ending point. The length of an effect chain depends on the considered level of granularity, which is determined through vertical traceability. Some approaches include different levels of granularity (Broy et al. 2011), others focus on one specific level like the functional level (Alt 2012).
- **Model-based linkage:** For modeling an effect chain, the set of system elements and dependencies has to be connected and modeled within a tool. Horizontal and vertical dependencies of system elements are captured by using modeling languages like SysML (Schwede et al. 2019; Broy et al. 2011; Chamas and Paetzold 2018; Meyer et al. 2011).

Limitations are:

- **Predefined focus of effect analysis:** The described methods have a specific focus, for example the fulfilment of customer requirements (Chamas and Paetzold 2018), the traceability of requirements (Schmitt 2020), functional design (Cao et al. 2006), modular product structures (Hackl and Krause 2017; Schwede et al. 2019) or automotive components (Broy et al. 2011; Kramer and Münzenberger 2010; Feilhauer 2018). This leads to a limited application area if needed information are not available or approaches cannot be adapted to other domains.
- **Limitation of development artefacts:** The identified methods are limited to the representation of artefacts created within the development process. Therefore, organizational or environmental artefacts, which affect the system and lead to rising complexity, cannot be considered. One example are limitations to mechanical surfaces (Frei, 2000).
- **Discipline-specific modeling:** Most approaches introduce a method for modeling effect chains within a certain discipline like mechanics (Frei, 2000), software (Becker and Mubeen 2018; Kaiser et al. 2010; Song et al. 2010) or safety (Zhao, 2017). This leads to limitations

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

regarding the scope of horizontal traceability and the application to complex technical systems.

- **No assignment of activities to specific roles:** The approaches describe modeling activities without assigning them to specific roles (Schwede et al. 2019; Frei 2000; Chamas and Paetzold 2018; Lee et al. 2018). Especially for extensive effect chains, expertise of several people with different roles must be consolidated to ensure that all relevant experts are consulted and all dependencies between artefacts are identified. Without role assignments, there is a risk of inconsistent effect chains and missing responsibilities for maintaining the actuality.

4. Method for effect-chain modeling and analysis

Based on the results of the systematic literature analysis, the MECA-method is developed. The method (see Figure 3) overcomes the four major limitations of existing approaches as follows: First, a generic and flexible structure is used to connect the main characteristics of effect-chain modeling and analysis approaches and to enable tailoring. Second, the MECA-method adds an initial step to define the individual goal of application. Based on the goal, starting and ending point as well as elements and dependencies of the deliberate view to be modeled are determined. Third, the method can be used in different contexts without focusing on specific artefacts, disciplines or domains. This is achieved by a flexible structure, but also by using SysML as a modeling language, which is designed for complex technical systems. In a SysML profile, organizational or environmental artefacts as well as specific cause-effect relationships can be defined, using stereotypes to capture required information. Fourth, it is possible to define views and viewpoints to specify role-specific information. Therefore, the MECA-method helps developers to identify and model cause-effect relationships between artefacts within an executable model, which can be used to analyse the specific effects.

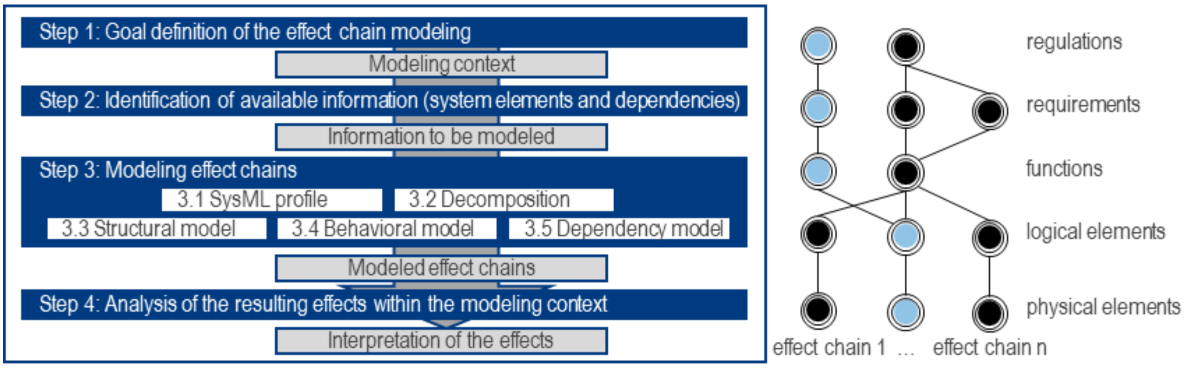

Figure 3. Model-based Effect-Chain Analysis (MECA) Method

Step 1: Goal definition of effect-chain modeling:

First, the goal of the effect-chain modeling must be defined. The considered modeling context determines the scope of required system elements and their cause-effect dependencies. The goal has to be clarified by typically involved MBSE roles, for example the Modeling Engineer or the Subject-Matter Expert (Gräßler et al. 2021). The goal has to be defined and available for all interacting persons to ensure the demand-oriented modeling. Exemplary goals for modeling effect chains are the fulfilment of external regulations or the evaluation of the effects of technical changes.

Step 2: Identification of available information

In the second step, dependencies of system elements within a delimited system section (horizontal scope) and over predefined system levels (vertical scope) have to be identified. System elements of different types are considered. The horizontal scope includes the determination of considered disciplines and the associated information from existing partial models. The vertical scope determines the granularity level of the effect chain, which has to be developed. Therefore, a set of modeling elements has to be defined.

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

The set can include typical development artefacts as well as organizational or environmental artefacts like regulations. In combination, all dimensions determine the information which is required to model the effect chain. The resulting output, the information to be modeled, serves as input for the next step.

Step 3: Modeling effect chains

The effect chain has to be modeled depending on the existing information. Compared to DSMs or DMMs, effect chain modeling enables the management of a high number of heterogeneous development artefacts (regulations, requirements, components, etc.) and different types of dependencies between them. Effect chains contain system element information (*nodes*) and their traceability information (*edges*) including the context specific cause-effect relationships. Initially, the modeling language is adapted by defining stereotypes in a profile (*step 3.1*). Stereotypes enable the mapping of information that is not specified by SysML, for example certification requirements as a kind of <<requirement>>. Additional, specific dependencies can be defined based on existing dependencies like <<allocation>>. Afterwards, all standardized and customized information can be modeled within the effect chain. Depending on the defined information set, the system is decomposed in different ways (*step 3.2*). These include the decomposition of requirements, functions, logical elements and physical elements. Based on the decomposition, a set of structure-oriented (*step 3.3*) and behavioral-oriented (*step 3.4*) diagrams is determined. These include the structure diagrams (block definition diagram, internal block diagram, package diagram, parameter diagram) and the behavior diagrams (activity diagram, sequence diagram, state machine diagram and use case diagram) of SysML. In the MECA-Method, the IPO logic (Alt 2012) within an internal block diagram serves as central functional part of the effect chain. With the help of the specific SysML profile any element can be allocated to this diagram. In the last step (*step 3.5*), the standardized and customized dependencies between these model elements are modeled, which result in a traceable effect chain between all system elements within the scope. It is recommended to use <<allocation>>, which can be adjusted by stereotypes.

Step 4: Analysis of the resulting effects within the modeling context

The effect chains can be analyzed by different approaches, for example impact analysis in context of engineering change management (Gräßler and Wiechel 2021; Hamraz et al. 2013) or requirement change management (Jayatilleke and Lai 2018). Also, existing approaches can be adapted to MECA-method and to the company-specific demands. Overall, the estimation of the effect is characterized by the degree of automation (non-automated, partially automated, fully automated); the traceability (not traceable, partially traceable, traceable) and the reference point of the estimation (quality, cost, time) (Gräßler and Wiechel 2021). The analysis depends on the goal of the effect-chain modeling (*step 1*), the available information (*step 2*) and the resulting information model (*step 3*). The systematic approaches help deriving qualitative or quantitative results with the help of algorithms or expert-knowledge.

5. Evaluation

Evaluation of the method is structured based on Blessing and Chakrabarti (Blessing and Chakrabarti 2009) and differentiates support evaluation and application evaluation. Support evaluation is conducted in parallel to method development and investigates whether the MECA-method is consistent and executable. Application evaluation is performed after the method development and investigates whether the MECA-method can be used for effect-chain modeling and analysis. Evaluation was carried out in the context of a project on effect-chain modeling to fulfill the UN ECE 156 for 30 systems of an automobile. The UN ECE (United Nations Economic Commission of Europe) is a European Commission which defines interdisciplinary regulations to regulate the europe-wide homologation of automobile series. Within the project, 36 workshops were executed with 25 experts (system managers and certification experts) for data collection to model effect chains. For validation of completeness and correctness of these effect chains, 30 further workshops were executed with the same 25 experts and additional 2 experts proving the correctness of the defined modeling method.
For the purpose of support evaluation of the generic MECA-method, a case example is conducted. The MECA-method is applied in effect-chain modeling for automotive parts. The example is developed

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

based on an effect-chain modeling project with a major German automotive company, which is confronted with a certification challenge. Figure 4 depicts the validation process for the MECA-method.

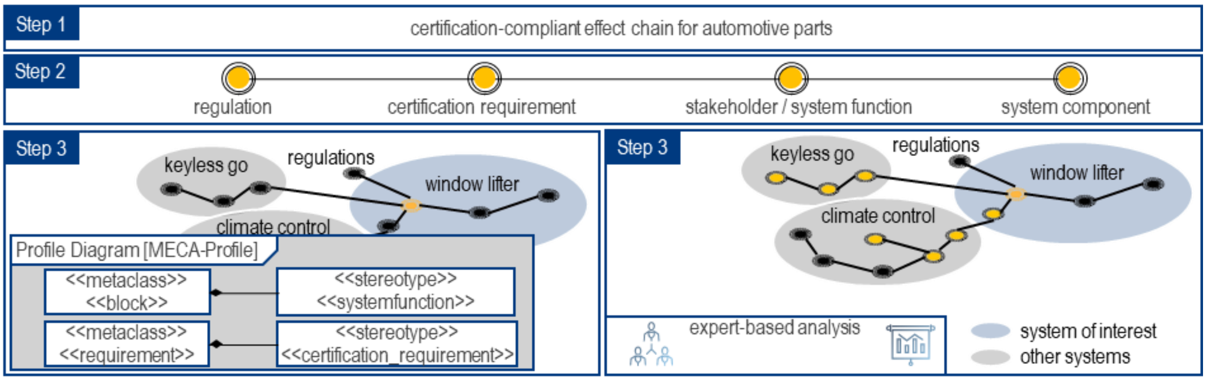

Figure 4. Validation process of effect-chain modeling (steps 1 to 4)

Step 1: Goal definition of effect-chain modeling

In the first step, the goal of the effect chain and the modeling context is defined. In automotive industries, several norms and regulations have to be fulfilled to obtain approval for a series of automobiles. One example is UN ECE 156 (UNECE 2021). The MECA-method is used to support the MBSE roles Modeling Engineer, Technical Manager and Subject-Matter Expert to model and analyze the required effect chains of the considered system and the depending systems. Therefore, the goal of the effect-chain modeling is to show the fulfilment of the UN ECE 156 systematically. The system under consideration is a *window lifter*. The window lifter is used for steering the windows and blinds of a car in parallel mode or as individual parts and can be assigned to several system levels. The overall system *car* can be divided into sub-systems such as powertrain, interior body and exterior body. The window lifter is located in the sub-system interior body as well as other systems like *smart key*. The window lifter moves when an input signal is emitted by the end user using *smart key* device. Also, the energy of *onboard power supply* is an input function for window lifter. Therefore, these three systems are part of the modeling context. A regulation which affects the window lifter is the UN ECE 21. The requirements from UN ECE 21 have to be met by all elements of *window lifter*, including software components such as sensors, actuators and electronic control units. The functions of the system can be decomposed within a functional hierarchy and are divided into *stakeholder functions* and technical *system functions*.

Step 2: Identification of available information

Based on the defined modeling context of step one, the following four elements to be modeled are defined: *Certification requirements* (1) derived from regulations, *stakeholder functions* (2), *system functions* (3) and *system components* (4). For identification of relevant information, workshops are executed with the system supervisor of the three *systems* and other persons, who are responsible for certification. Before the workshop, four different dependency matrices are prepared. In the rows and columns of each matrix, two relevant artefacts are assigned to each other. For example, one dependency matrix includes stakeholder functions (rows) and system functions (columns). Within the cells, different types of dependencies can be recorded. Therefore, in the workshop, the supervisor has to identify the dependencies between those two relevant elements by allocating them within the matrices. Based on that, an IPO-logic (input, processing, output) is used to model the effect chains in accordance to Alt (Alt 2012). Afterwards, interdependencies between *stakeholder* and *system functions* and *system components* (sensors, actuators and control units) are collected. For the window lifter, 6 stakeholder functions and 12 system functions can be identified. The functions are realized for 13 components such as the head unit or the body controller. System functions of the systems *smart key* and the *onboard power supply* are included in the *stakeholder function* of the *window lifter* and represent the interdependences between the systems. Knowing the functions and the components of

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

the window lifter, the mapping with *certification requirements* is executed. Based on the collection of information in step 2, the UN ECE R21, which contains certification requirements regarding anti-trap protection, can be mapped on *stakeholder functions*, *system functions* and *system components*.

Step 3: Modeling effect chains

Main information of the effect chain are modeled using a <<block>> for each certification requirement, system function and system component. These blocks can be categorized into an IPO-logic within an internal block diagram (ibd). The main part of the effect chain of the stakeholder function "automatic opening of the window" is shown in the middle of Figure 5. Each stakeholder function is represented by an own IPO-logic, where all internal and external system functions are categorized. Beside the system functions, certification requirements and system components can be linked to elements within the IPO-logic. To model interdependencies with the certification requirement, a separate block definition diagram (bdd) is created (Figure 5 left). Dependencies of certification requirements on stakeholder functions, system functions and components are modeled using <<allocate>> dependencies. In the last step, the components are mapped in a similar diagram as bdd.

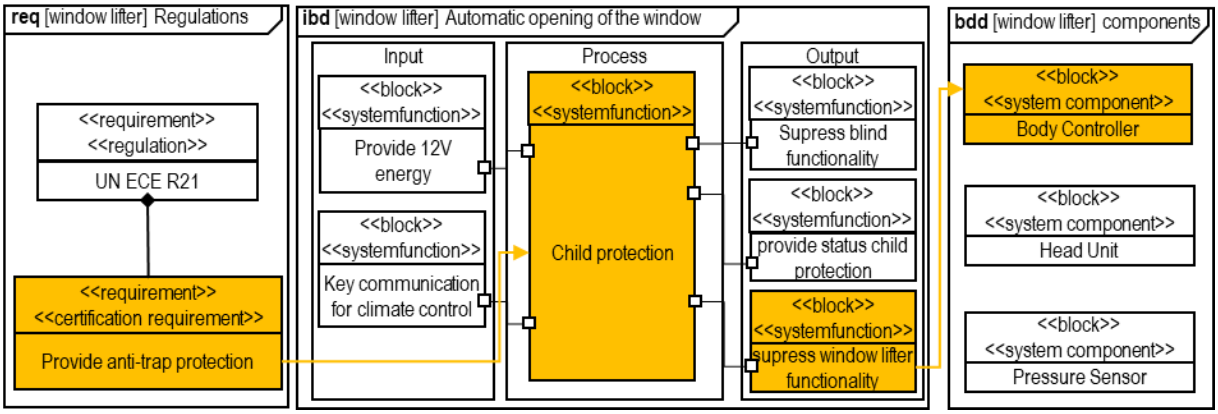

Figure 5. Effect chain (yellow nodes and edges) of function "automatic opening of window"

Step 4: Analysis of the resulting effects within the modeling context

Validation of the modeled effect chain is executed within separate workshops. The participants of the workshop are the system supervisor(s) and the persons in charge for the certification of the system.
First, the modeled effect chains are checked with regard to their completeness and correctness. To do so, it is checked whether all *stakeholder functions*, *system functions*, *system components* and *regulations* are included and if the dependencies between these elements are modeled correctly. For validation purposes, fictitious changes are used by the participants: a software update regarding the head unit of the overall system. It is checked whether the head unit depends on the regulation for anti-trap regulation (*UN ECE R21*), which is not the case. Since the head unit is used by the stakeholder function "automatic opening of the window", the update has also an effect on this stakeholder function. The stakeholder function "automatic opening of the window" includes the system function "Key communication" of the system *smart key*, which has a dependency on the regulation for keys. Therefore, an effect of a software update regarding the head unit has an effect on a dependent system. That effect could be systematically detected by the use of the MECA-method.

Application Evaluation

For application evaluation issues during effect-chain modeling were captured and used for improvement of the MECA-method. A finding was, that the goal and the user's contribution has to be communicated explicitly. This is why the definition of the overall goal (method step 1) was added as a main step. A second finding was, that during workshops it remained fuzzy which data is relevant for modeling effect chains. The application of the MECA-method indicates that the main time effort results from the data collection (42 % of time effort) and validation (23 % of time effort). The modeling itself is a minor

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

additional effort (35 % of time effort) of the time and can be reduced by models. As a consequence, within the MECA-method it is highlighted to perform goal specific tailoring in every step. This also improves efficiency of application, by focusing on necessary data. Overall, using the MECA-method for the case example demonstrates the applicability of the method in industrial context.

6. Conclusion and outlook

The research question is answered by defining the Model-based effect chain Analysis (MECA) method, which combines main characteristics of existing methods and overcomes their limitations. The method enables developers to systematically model effect chains and understand the interactions and propagation effects within the systems and its context. Based on the application of the MECA Method an executable system model for effect chain analysis can be created and used for simulations like impact analysis. Effect chains can also be expanded with conventional modeling methods like OOSEM (Friedenthal et al. 2008) towards a holistic system model. The MECA-method can be understood as a possible introduction to implement model-based development in industry. The MECA-method can be used to model a deliberate view of a system model, including a flexible starting and ending point, adjustable to the individual goal. Additionally, company-specific artefacts can be included by adapting the SysML profile, which enables the model-based linkage of the artefacts as well as the consideration of roles. To foster understanding of the method, a case example is given: modeling and analyzing certification-compliant effect chains for the window lifter of an automotive vehicle. Evaluation based on 30 automotive systems, 66 workshops and 27 industry experts prove applicability of the method. In further research, the approach will be validated in different domains, which pursuit other goals (e.g. impact analysis for engineering changes). In addition, the analysis of the method is to be automated using algorithms and evaluation metrics to infer the required systematic results. The application of the MECA-method helps to perform development of complex technical systems according to common certifications but also to reduce negative impact from unrecognized changes and project failure.

References

Albers, A.; Zingel, C. (2013): "*Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML*." In: Michael Abramovici and Rainer Stark (Hg.): Smart Product Engineering. Berlin, Heidelberg, pp. 83–92. https://doi.org/10.1007/978-3-642-30817-8_9

Alt, O. (2012*): Modell-basierte Systementwicklung mit SysML. In der Praxis*. Hanser, Carl, München.

Becker, M.; Mubeen, S. (2018 - 2018): Timing Analysis Driven Design-Space Exploration of Cause-Effect Chains in Automotive Systems. In: *IECON* 2018: IEEE, pp. 4090–4095.

Blessing, L. T. M.; Chakrabarti, A. (2009): *DRM, a Design Research Methodology*. Vol. 1. Springer, Surrey.

Broy, M.; Reichart, G.; Rothhardt, L. (2011): Architekturen softwarebasierter Funktionen im Fahrzeug: von den Anforderungen zur Umsetzung. In: *Informatik Spektrum* Vol 34 No. 1, pp. 42–59.

Cao, G.; Guo, H.; Tan, R. (2006): Effect and Effect Chain in Functional Design. In: Kesheng Wang, George L. Kovacs, Michael Wozny and Minglun Fang (Hg.*): Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, Vol. 207: Springer US, pp. 412–420.

Chamas, M. W.; Paetzold, K. (2018): Modeling of Requirement-Based Effect Chains of Mechatronic Systems in Conceptual Stage. In: *IJEETC*, Vol. 7 No. 3, pp. 127-134. https://doi.org/10.18178/ijeetc.7.3.127-134

Clarkson, P. J.; Simons, C.; Eckert, C. (2004): Predicting Change Propagation in Complex Design. In: *J. Mech. Des.* Vol 126 No. 5, pp. 788–797. https://doi.org/10.1115/1.1765117

Danilovic, M.; Browning, T. R. (2007): Managing complex product development projects with design structure matrices and domain mapping matrices. In: *International Journal of Project Management* Vol 25 No. 3

Delligatti, L. (2014): *SysML distilled. A brief guide to the systems modeling language*. Addison-Wesley, NJ.

Dobrusskin, C. (2016): On the Identification of Contradictions Using Cause Effect Chain Analysis. In: *Procedia CIRP* 39, pp. 221–224. https://doi.org/10.1016/j.procir.2016.01.192

Eger, T.; Eckert, C. M.; Clarkson, P. J. (2007): Engineering Change Analysis during Ongoing Product Development. In*: ICED (Hg.): DS 42: Proceedings of ICED* 2007, ICED, Paris, France.

Estefan, J. A. (2008): Survey of model-based systems engineering (MBSE) methodologies. Focus Group 25.8. INCOSE MBSE initiative.

Feilhauer, M. (2018): *Simulationsgestützte Absicherung von Fahrerassistenzsystemen*. University Stuttgart.

Frei, N. (2000): Programm zur Auslegung mechanischer Wirkketten. In: *DFX 2000: Proceedings of the 11th Symposium on Design for X.* Schnaittach/Erlangen, S. 71–76.

https://doi.org/10.1017/pds.2022.191 Published online by Cambridge University Press

Friedenthal, Sa.; Moore, A.; Steiner, R. (2008): *A practical guide to SysML. The systems modeling language*. Burlington, Mass.: Elsevier/Morgan Kaufmann. https://doi.org/10.1016/C2013-0-14457-1

Gräßler, I. (2015): Implementation-oriented synthesis of mechatronic reference models. In: *Torsten Bertram (Hg.): Fachtagung Mechatronik 2015. VDI Mechatronik. Dortmund*, 12.03.-13.03. Aachen: Institut für Getriebetechnik and Maschinendynamik, pp. 167–172. https://doi.org/10.31224/osf.io/xspbm

Gräßler, I.; Oleff, C.; Preuß, D. (2022): Proactive Management of Requirement Changes in the Development of Complex Technical Systems. In: *Applied Sciences* Vol 12 No. 4, pp. 1874.

Gräßler, I.; Wiechel, D. (2021): Systematische Bewertung von Auswirkungsanalysen des Engineering Change Managements. In: *Proceedings of the 32nd Symposium Design for X (DFX2021)*.

Gräßler, I.; Wiechel, D.; Pottebaum, J. (2021): Role model of model-based systems engineering application. In: *IOP Conf. Ser.: Mater. Sci. Eng.* 1097 (1), p. 12003.

Hackl, J.; Krause, D. (2017): Towards an impact model of modular product structures. In: DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design, Vancouver, Canada, 21-25.08.2017, pp. 151–160.

Hamraz, B. (2013): *Engineering change modelling using a function-behaviour-structure scheme*. Apollo - University of Cambridge Repository. https://doi.org/10.17863/CAM.14061

Hamraz, B.; Caldwell, N. H. M.; Clarkson, P. J. (2013): A Holistic Categorization Framework for Literature on Engineering Change Management. In: *Syst. Engin.* Vol 16 No. 4, pp. 473–505.

Hick, H.; Bajzek, M.; Faustmann, C. (2019): Definition of a system model for model-based development. In: *SN Appl. Sci.* Vol 1 No. 9. https://doi.org/10.1007/s42452-019-1069-0

International Council on Systems Engineering (2007): *Systems Engineering Vision* 2020. INCOSE.

Jayatilleke, S.; Lai, R. (2018): A systematic review of requirements change management. In: *Information and Software Technology* 93, pp. 163–185. https://doi.org/10.1016/j.infsof.2017.09.004

Kaiser, B.; Klaas, V.; Schulz, S.; Herbst, C.; Lascych, P. (2010): Integrating System Modelling with Safety Activities. In: David Hutchison et al. (Hg.): *Computer Safety, Reliability, and Security*, Bd. 6351. Springer, Berlin, Heidelberg. pp. 452–465. https://doi.org/10.1007/978-3-642-15651-9_33

Kirova, V.; Kirby, N.; Kothari, D.; Childress, Glenda (2008): Effective requirements traceability: Models, tools, and practices. In: *Bell Labs Tech. J.* Vol. 12 No. 4, pp. 143–157. https://doi.org/10.1002/bltj.20272.

Kramer, T.; Münzenberger, R. (2010): Modellierung and Echtzeitanalyse komplexer Wirkketten in Fahrerassistenzsystemen. In*: 3. AutoTest*, FKFS, Stuttgart.

Lee, M.-G,; Chechurin, L.; Lenyashin, V. (2018): Introduction to cause-effect chain analysis plus with an application in solving manufacturing problems. In: *Int J Adv Manuf Technol* 99 (9-12), pp. 2159–2169.

Machi, L. A.; McEvoy, B. T. (2012): *The literature review*. 2. ed. Corwin, Thousand Oaks, Calif.

Meyer, J.; Holtmann, J.; Meyer, M. (2011): Formalisierung von Anforderungen and Betriebssystemeigenschaften zur frühzeitigen Simulation von eingebetteten, automobilen Systemen. In: *8. Paderborner Workshop Entwurf mechatronischer Systeme*, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Vol. 294, Paderborn.

Object Management Group (2004): OMG: Unified Modeling Language Infrastructure Specification, Version 2.0.

Object Management Group (2019): OMG Systems Modeling Language (OMG SysML™).

Schmitt, Nicholas (2020): *Durchgängiges Vorgehensmodell zur Anforderungserfassung für die Entwicklung mechatronischer Systeme im Automobil*. UB-PAD - Paderborn University Library.

Schwede, L.-N.; Hanna, M.; Wortmann, N.; Krause, D. (2019): Consistent Modelling of the Impact Model of Modular Product Structures with Linking Boundary Conditions in SysML. In: *Proceedings of the Design Society International Conference on Engineering Design* Vol. 1 No. 1, pp. 3601–3610.

Song, M.; Yin, G.; Wang, H.; Ni, J. (2010 - 2010): Effect-Oriented Function Analysis and Testing Method. In: 2010 *Fifth International Conference on Internet Computing for Science and Engineering*. 2010 (ICICSE). Harbin, China, 01.11.2010 - 02.11.2010: IEEE, pp. 34–38. https://doi.org/10.1109/ICICSE.2010.37

UNECE (2021): UN Regulation No. 156 - Software update and software update management system.

Guideline VDI 2221:2019 (2019): VDI 2221 Design of technical products and systems.

Walden, D. D.; Roedler, G. J.; Forsberg, K.; Hamelin, R. D.; Shortell, T. M. (2015): *Systems engineering handbook*. Vol. 4. Wiley, Hoboken, New Jersey.

Weilkiens, T. (2014): *Systems Engineering with SysML/UML*. dpunkt, Heidelberg.

Zhao, X. (2017): The Integration of Model-Based Safety Analysis and Model-Based Systems Engineering at an Early Stage. In: *JSST* Vol. 10 No. 1. https://doi.org/10.13034/jsst.v10i1.163