INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.203

Towards Integrated Safety Analysis as Part of Traceable Model-Based Systems Engineering

M. L. Valdivia Dabringer ✉, A. Dybov, C. Fresemann and R. Stark

Technische Universität Berlin, Germany

✉ valdiviadabringer@tu-berlin.de

Abstract

Currently systems grow in complexity and more aspects, such as socio-technical aspects or the obligation to produce proof become more important. Both require a safety analysis on the system level early in the design process. System overview is provided by MBSE, while safety analysis is provided for example by FMEA. Both processes are executed organizationally and timely separated from each other. This research proposes a concept and a tool integration at the concept design phase, during system definition and functional decomposition and evaluates the effect and its potential applicability.

Keywords: model-based systems engineering (MBSE), safety analysis, traceability

1. Introduction

Systems Engineering provides the opportunity to connect and complement various models along the development cycle. This enables an overview of the technical system content itself, of potential implications e.g. on safety aspects as well as of tests and conformity proofs. This research follows the INCOSE understanding of Systems Engineering (Sillitto *et al.*, 2019) as a transdisciplinary and integrative approach where a system may be composed of people, services, processes, natural elements, etc. and it is considered as interacting with its operational environment. Model Based Systems Engineering (MBSE) (Estefan, 2008) approaches often refer to the V-Model (INCOSE, 2015; Bender, 2005) as the underlying development process. A connecting point to the Safety Analysis Processes is considered by the authors during the system definition, decomposition and analysis process in the left-hand side of the V-Model, as indicated by the red box in Figure 1. Road Safety is defined in the ISO 26262-1 (ISO, 2018) as the absence of non-necessary risks. Engineers often apply FMEAs or FMECAs in order to investigate on potential risks compromising the safe operation of components or systems. Systems Engineering tools such as the Cameo Systems Modeller or Capella provide already an integration to safety or reliability data as proposed for aeronautic use cases (Mhenni *et al.*, 2016), or even described by the OMG working group and tool vendors (Biggs *et al.*, 2018). However, the integration on a process level within the V-Model or as a method covering the approach from the modelling aspect including the traceability until the system's validation is currently missing. MBSE is evolving towards the direction of Advanced Systems Engineering (ASE) (Albers and Lohmeyer, 2012), considering especially socio-technical aspects such as human behaviour and subjective interpretation of a system.

The intent of this paper is to describe a model-based method that considers functional safety and socio-technical aspects during the early stage design processes.

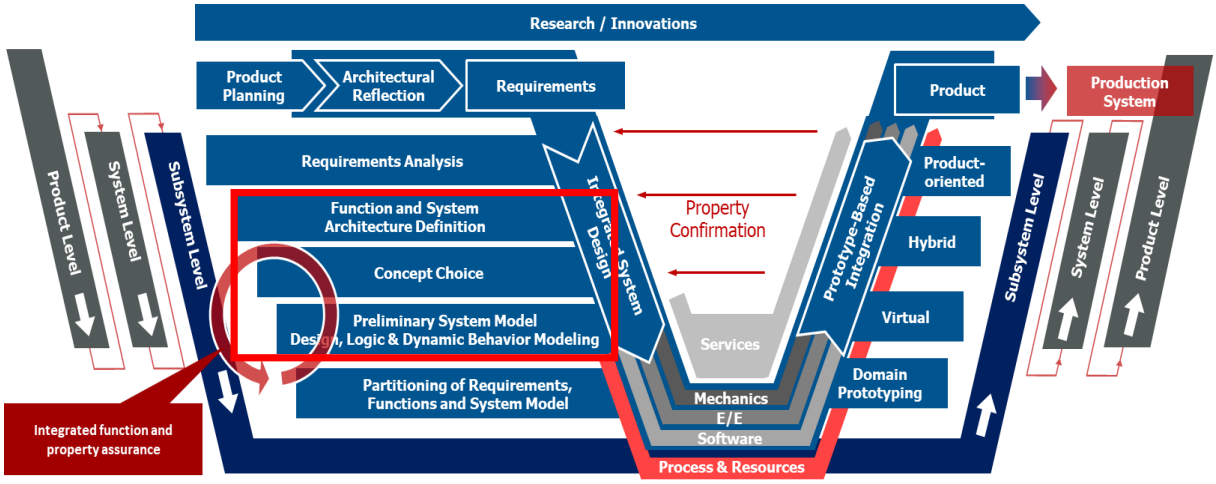

Figure 1. V-Model, according to (Bucholz *et al.*, 2018)

2. State of the Art

Research has already been made in the direction of integrating safety analysis to the systems engineering process. Proposals include the automatic generation of safety analysis parting from a SysML model. The existing solutions part of using plug-ins to extend the System Model and generate safety analysis automatically, or extensions of the SysML Model by adding FMEA capabilities to it.

Riel et al. (2018), propose a methodology to integrate functional safety and cybersecurity aspects in the early design phases by a systematic approach to integrated requirements elicitation based on axiomatic design and signal flow analysis. This is achieved through the decomposition of a high-level functional safety requirement defined by a hazard and risk analysis to system safety functional requirements and then further to technical safety concept level safety functional requirements. The signal-flow analysis enables a systematic approach of the process to find potential sources of violation of the safety goals that leads to the breaking down of the high-level requirements. Afterwards, the definition of design parameters during the technical safety design leads to additional technical safety concept level functional safety requirements. This methodology is proposed by the authors as an extension of Automotive Spice.

Helle (2012) proposes a method to extend the existing meta model with concepts from the safety domain. Functions are designed in per activity diagram in SysML and are allocated to system component, modelled as blocks, that are extended by an additional stereotype to check its safety relevance. Failure cases are defined as use cases marked with a specific stereotype and a maximum allowed failure rate and are dependency associated with a specific stereotype to a function. With support of a Java program called SafetyAnalyzer, using the defined stereotypes in the system models, its elements are converted into corresponding internal data types. As an output the user gets a minimal cut set for each failure case and system alternative, a reliability block diagram representing this minimal cut set graphically, a safety verdict for each failure case and an overall safety verdict.

David et al. (2010) propose a solution to generate automatically an FMEA from structural and behavioral diagrams in SysML Models called MéDISIS. To achieve its purpose, the Internal Block Diagram, the Block Definition diagram, the Activity Diagram and the Sequence Diagram of the System Model derive to a list of failure modes and their possible causes and effects for each component and function. The final FMEA is then complemented by a safety expert. Finally, MéDISIS maps the SysML model to the AltaRica language (Arnold *et al.*, 1999), in which the FMEA is performed, to correctly assign the generated failure modes generated in the first step.

Another proposal is a model-centric format performed by a plug-in developed to connect the FMEA information to the SysML Model directly (Biggs *et al.*, 2018). This plug-in creates a new UML class into the Tool called FMEAItem, with the characteristics of defining the severity, the occurrence and the detectability within the same tool. Based on these, the risk priority number (RPN) can be defined and assigned to the corresponding functions or components in SysML. This RPN is then compared to

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

a threshold defined by the company, so that the impact each FMEAItem can be defined within the company standard. This gives new information to take decisions such as the redesigning of the relevant systems or a redefinition of requirements. Afterwards, the FMEAItem can be updated with the new information to calculate a new RPN.
The presented methodologies have coincidences in extending the SysML Model to add new safety information to it. Our proposed approach includes the connection of an analysis performed in an external tool like MS Excel to the systems engineering process. MS Excel is not a vendor-made tool to perform an FMEA analysis, however it fulfils the features needed to perform such an analysis. It includes the storage of different type of data in a tabular format and the ability to multiply numerical values. This is why this tool was chosen to perform the FMEA analysis.
The proposed methodology allows experts to perform their analysis in external tools and have a traceability among the different models to have a better understanding of the process.

3. Solution Proposal

In order to provide a methodical and technical solution, this research establishes links between virtual and hybrid prototyping elements as presented in Figure 2. A test person will interact with the hybrid prototype at the point in time when an object to test is functionally defined early in the product development cycle. This enables to test the system (Bucholz *et al.*, 2018), and at the same time to test the socio-technical behaviour including the human-machine interaction and identify potential risks. Furthermore, a BDD and an Interaction Structure will be coupled with a component and system risk assessment, assessing the usefulness of the interaction between systems with respect to overview, handling and benefit from systems overview as well as changes occurring to one or the other data-set.

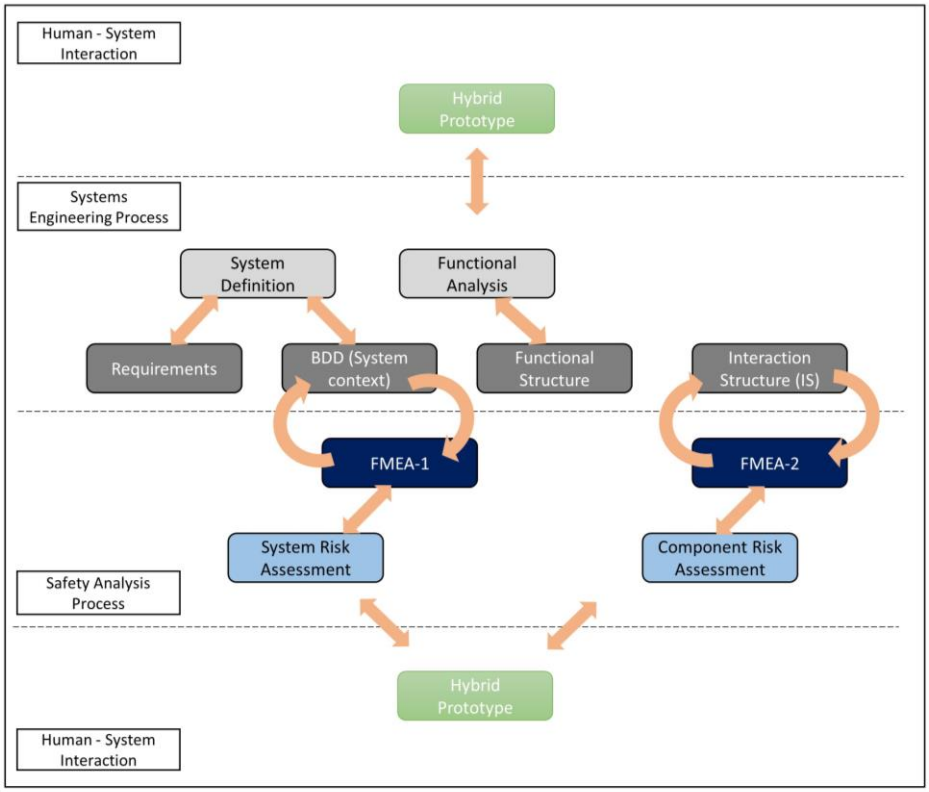

Figure 2. Model Coupling of Systems Engineering and Safety Analysis Processes

The connection of the criticality, resulting from the safety analysis (FMEA), the system model (Cameo System Modeler) and the behaviour model (MATLAB Simulink) is performed in three steps, depicted as red arrow in Figure 3. The linkage of the FMEA to the requirements, the linkage of the requirements to the system model and the linkage of the system model to the behaviour model. In the first two steps the

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

capabilities of the modelling tools were used to establish the linkage, while the third step is performed by Intercax Syndeia. Syndeia is an add-on to the Cameo System Modeler and supports connections on model, object or element level between the Cameo System Modeler and MATLAB Simulink.

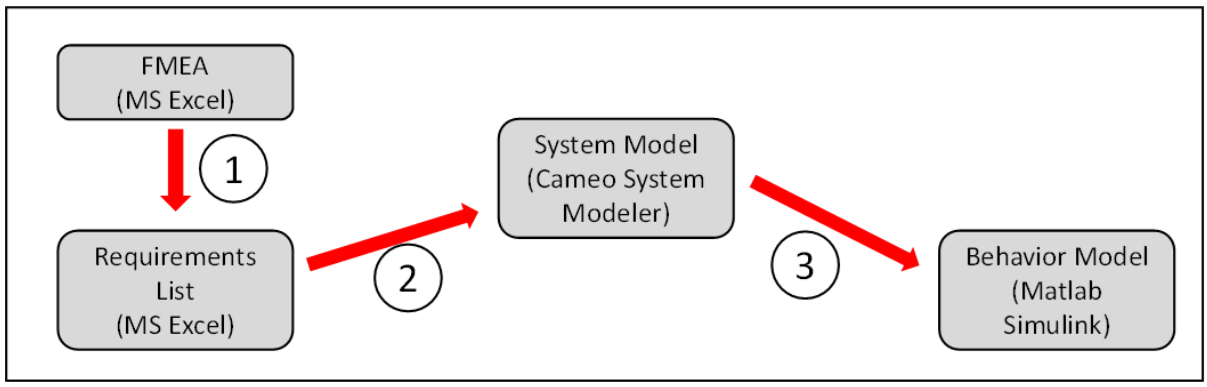

Figure 3. Path to connect a safety analysis with systems engineering methodology

3.1. Linkage between FMEA and Requirements List

The first step to link the results obtained from the FMEA to the requirements list is to add new requirements to that list, that include the corresponding criticality. Then that requirement is linked through the hyperlink function of MS Excel to the criticality in the FMEA workbook. Excel does not support modification to the hyperlinks. When the hyperlink is successfully established, the text in the requirement Excel cell turns blue and underlined as depicted in Figure 4, and following the link forth and back is supported. The change management limitation of this approach, might be solved by a macro, and the need to set a fix hyperlink. However, since the FMEA is a static model, this is not a critical limitation, and the purpose of tracing back the requirements to the FMEA is fulfilled.

Nr	Category	Type	Description	Derived From
8	FMEA	Non-functional	The system has the assigned hazards of each components	
8.1	FMEA	Non-functional	The stepper motor has a criticality of 35 when having a mechanical failure that blocks the rotor.	8
8.2	FMEA	Non-functional	The potentiometer has a criticality of 28 when decalibrating and sending a wrong value to the microcontroller	8
8.3	FMEA	Non-functional	The microcontroller's software has a criticality of 14 when having a I	8
8.4	FMEA	Non-functional	The powe supply pc	
8.5	FMEA	Non-functional	The alert corrossio	
8.6	FMEA	Non-functional	The alert electrical	
8.7	FMEA	Non-functional	The micr having a I comman	
8.8	FMEA	Non-functional	The contr electrical	
8.9	FMEA	Non-functional	The signa sending t	
8.10	FMEA	Non-functional	The alert circuit th	
8.11	FMEA	Non-functional	The poter power ov	
8.12	FMEA	Non-functional	The stepp functiona	

Description	Severity (S)	Probability of occurence (O)	Detectability (D)	Risk priority number (RPN)	Criticality
Mechanical failure of the stepper motor, that blocks the rotor	7	5	4	140	35
Decalibration of the Potentiometer, sending a wrong signal to the Microcontroller	7	4	8	224	28
Bug in the Microcontroller's software that executes the wrong command	7	2	8	112	14
The power supply does not provide power to the alert system	5	10	2	100	50
Failure of alert system due to corrosion	5	3	8	120	15
Failure of alert system due to electrical overload	5	7	7	245	35

Figure 4. Linkage between FMEA and Requirements List

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

3.2. Linkage between Requirements List and System Model

The Cameo System Modeler has the option to manage a requirement list locally in the tool. This enables a traceability between the requirements and the system architecture. The challenge with this method is synchronizing the external requirements list to the tool internal requirements list. The Cameo System Modeler has the capability of synchronizing its own requirements list with an MS Excel. For the optimal synchronization, the headers in Excel should be the same as the headers in the Cameo System Modeler. The ID of each requirement must be a non-numerical value. The synchronization of both models is not made on real time; however, the current state of the model can be copied bidirectionally. This represents a risk of overwriting the data in the originally planned Requirements List tool, since for design purposes it will be performed on MS Excel and not inside the Cameo System Modeler. When the requirements are changed in the MS Excel Model. The Cameo System Modeler informs of the changes giving a visual alert, by highlighting the requirement blue, green or red for a modified requirement, a new requirement or a deleted requirement correspondingly.
The linkage between the requirements and the components in the system architecture are performed within Cameo System Modeler in the traceability matrix. This allows to assign the requirements to the corresponding system architecture artefact. By the linkage of the FMEA requirements to the components, the hazards calculated in the FMEA are linked to the component.

3.3. Linkage between System Model and Behavior Model

Establishing a traceability from the safety analysis until the systems engineering process, the hazards assigned to the components in the system model have to be linked to the behavior model. Neither the Cameo System Modeler nor MATLAB Simulink support the linkage between artefacts among the models. This leads to a third-party solution called Syndeia, that enables the linkage between the corresponding artefacts. In this example, Syndeia works as a Cameo System Modeler plug-in. Syndeia can work through a variety of open standards, open-source projects and libraries, and production ready APIs. The linkage for this example is technically solved through an API. Once the plug-in is installed, both models are shown in the Synedia's internal connection manager. The links are created by drag and dropping the elements to be linked on each other. Once the connection is performed, the corresponding linked elements can be opened either from the SysML model itself, or through the connection manager.

3.4. Overall integration of Safety Analysis with the System Engineering Process

Through the linkage of artefacts mentioned above, a connection through the systems engineering process can be done indirectly by the process and following the steps. The linkages enable the traceability, so that an element in the behavior model can be traced back to a safety analysis element by following backwards the system engineering process.

4. Use Case: Safe Door Opening System (SDOS)

A test bench for a hybrid Safe Door Opening System (SDOS) was developed (Figure 5). This system is conceived as driver assistant system for cars giving a warning whenever a dooring accident might occur. The SDOS will generate different warnings informing the driver whenever an incident might occur. The SDOS (Figure 5, left) consists of two warning light systems for the driver and cyclist, a vibration signalinside the door handle, a sound warning and a motor to close the door or stop the door in a safe position. The warning signals will be presented to the driver depending on the door opening angle (Figure 5, right) combined with signals from a virtual surrounding (e. g. cyclist approaching very quickly from behind).Therefore, the developed test stand consists of a functional prototype of the door, the simulation software dSpace visualising the car as well as the surrounding including other participants and MATLAB, where the warning signals are generated as a response to virtual sensors detecting the cyclist.
This test stand allows to evaluate the integration of various models or parameters and their relationship with each other. Thus, it allows to trace the model-chain from requirements to implementation in the system model and functional prototype. Furthermore, it allows human-in-the-loop tests during the

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

conceptual development phase, e.g. for a comfort check for the driver or accident such as catching one's foot in the door because this closes automatically. Socio-technical interaction between a cyclist and a driver may be tested during the passing. For example, it would be realistic that the cyclist performs a stop manoeuvre when he or she recognizes the door opening while the driver is stuck in the car- or both continuing their activities and, in that moment, causing and accident.

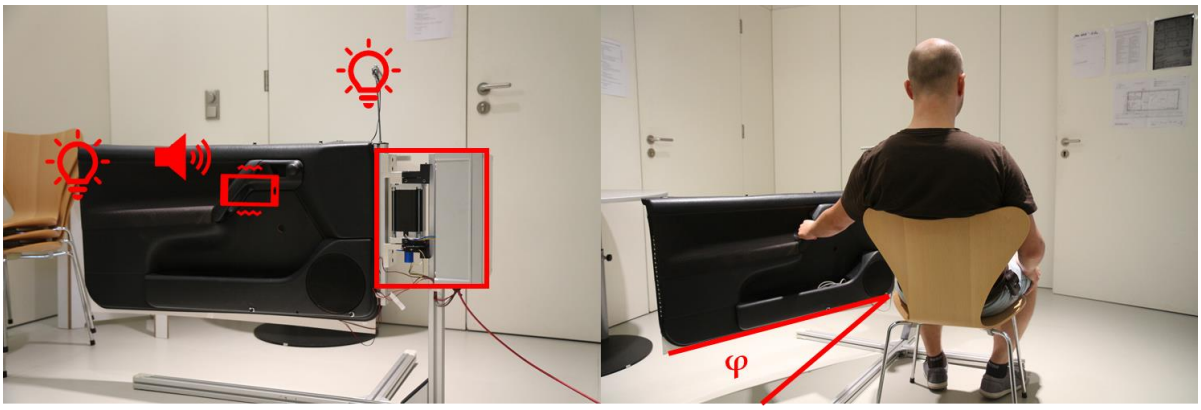

Figure 5. Experimental set-up of SDOS (left); Experimental set-up with test person, manual test, φ - opening angle (right)

The testing of the system, its functionality and behavior is done by a prototype where carious project models are linked together. This process is located at the bottom and on the right branch of the V-Model (Figure 1). According to the steps of the right branch, a prototype was assembled, with the help of which the system units, individual system elements and the necessary functionality of the process of opening and closing the door are tested. Then, the behavior model in conjunction with the mechanical part of the prototype is tested. The IEEE 829-2008 standard was used to plan and describe system tests. Thus, each individual process was represented in separate actions and tasks, which were reproduced using prototype hardware and behavior models. This standard was chosen for its similarity to the V-Model and its approach of component testing to full system testing.

Three testing scenarios were selected for full testing and analysis of the safety of units, elements, functions and requirements. These scenarios cover the range of possible situations of behaviours that were defined in the requirements and test the SDOS for each of them. In each scenario a step by step testing of units, components, subsystems, systems and the prototype as a whole is conducted. At each step, it is possible to trace the relationships of the models and their transmission of signals and information. With the help of MBSE methods, in particular V-Model, on the example of a functional prototype, aspects of the integration of models and their relationships with each other are studied.

Experimental prototype made it possible to connect several technical devices and test them together. First of all, the prototype connects different notification systems (illumination system, haptic system, sound system) for the driver and passenger. Secondly, the prototype has an active system of blocking and moving the door, which can block the door in the event of a threat or return it to a safe position. Thirdly, the prototype brings the different design models together: requirements, FMEA, functions, system elements, and behavior. In this case, all models are connected together and it becomes possible to trace the relationships and potential problems.

The purpose of each scenario is to examine the behavior of the system, test the operation of individual units and system components, use behaviour models to understand how the failure of one system part leads to a change in the functionality of the entire system and to test the security and response of the system to a changing situation.

Application Scenario 1:

The first application scenario describes a closed door (φ=0 - degree angle) and an approaching cyclist from behind. When the cyclist approaches to a distance between 10 and 15 meters, the front LED on

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

the back mirrors lighten up. At a distance between 5 and 10 meters, the vibration motor in the door handle and the front LED are activated. At a distance of less than 5 meters between the cyclist and the car door, the front LED and the driver alert sounds are activated.

Application Scenario 2:

The second application scenario describes a situation with the cyclist approaching from behind, but with the ajar door, an angle from φ = 1 to 17 degrees. In this case, the front and rear LEDs light up at a distance of 10 to 15 meters. In addition, the vibrating motor is immediately activated in the door handle. At a distance of 5 to 10 meters, the driver's audible signal is activated. When the distance between the cyclist and the car is less than 5 meters, the front and rear LEDs light up, the vibroacoustic alarm is activated and further door opening is blocked.

Application Scenario 3:

The third use case describes a situation with a wide-open door (door opening angle (φ) larger than 17 degrees, Figure 5, right), a cyclist approaching from behind. At a distance of the cyclist between 10 and 15 meters, the front and rear LEDs light up. A vibration motor is activated at a distance between 5 and 10 meters. If the distance between the cyclist and the car door is less than 5 meters, an audible alarm will sound. In addition, the door closes to an angle of 17 degrees, due to the internal motor, and prevents or makes it difficult to re-open further until the cyclist passes.

5. Evaluation and Conclusions

First of all, we could demonstrate traceability between the corresponding design artefacts from the requirements until the tests investigating potential risks as well as the socio-technical implications. This allows engineers to follow the information chain during the design process, to support the verification of the fulfilment of the requirements and to apply change management in parallel.
When applying the three testing modes, several functional chains of the product were tested, thereby revealing direct connections between all elements and models. Furthermore, the practical implementation showed how specific calculations of possible incidents affect the operation and the choice of solution elements. Figure 6 (red words and boxes) shows the relationships between the different error levels and what they can lead to, such as engine failure or signal breakdown.

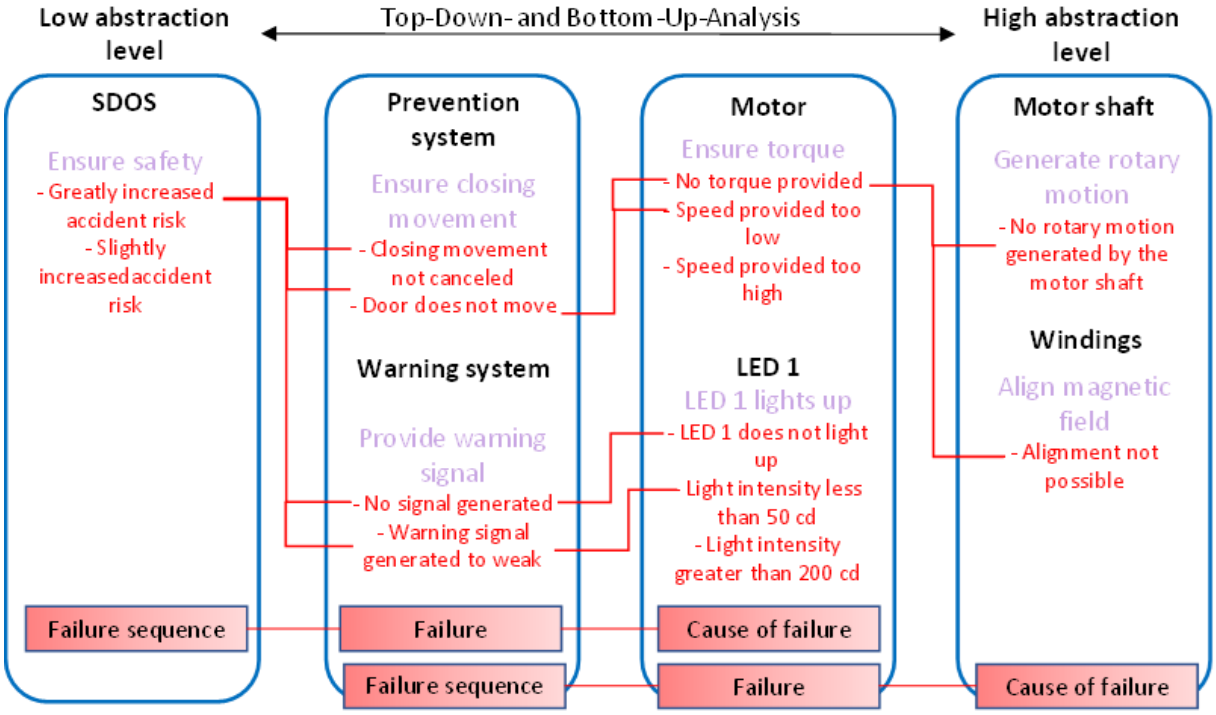

Figure 6. Failure analysis across system elements (black words) and functions (purple words)

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

The Top-Down approach enables performing an analysis breaking down the system into subsystems and if required to component or function level. Then a functional analysis is performed using the function tree and FAST-Diagram according to VDI 2803 standard. Figures 6 and 7 represent function trees going from a higher to a lower level of abstraction with an increasing level of detail. The chain of functions across different levels of detail, clarifies how the function is implemented in increasing detail direction and explains the reason of existing of the subsequent function in decreasing detail direction. The FAST functional structure gives a logical path in a continuous chain from higher-level functions through the base and subsequent functions to the adopted function. (Verein Deutscher Ingenieure, 2019) First, the functions of the corresponding elements of the system are determined. For example, SDOS has a function called "Ensure safety" (Figure 7, purple) and the prevention system has a function called "Ensure closing movement". It is convenient to start the linkage top-down, remaining at first on a higher granularity level. Once all functions are linked to risks, subfunctions may be connected. During the third step of the error analysis task, previously defined functions are associated with potential errors to create a so-called network of failures. After that, it becomes possible to find the consequences and the reason for the refusal (Figure 7). Thus, a full-fledged model of connections between models and potential incidents for the product and its user has been established. In that manner, a higher traceability and visibility is supported during the product development and testing.

During the research the first limitations occurred by the use of the tools themselves, since MS Excel is not the optimal tool for performing a requirements list or a FMEA. There were also limitations related to the cross-tool traceability, since the solution applied is a combination of tool-supported linkage and add-on supported linkage. This breaks the information chain, as it is necessary to follow the complete design path to trace back to an initial artefact (e.g. an FMEA). Trace links are applied manually inside the Syndeia tool by dragging and dropping the elements to be linked within the tool. This manual work can have an impact on the acceptance of the traceability tool, since engineers could consider that manually linking elements together is time consuming and unprofitable for their work.

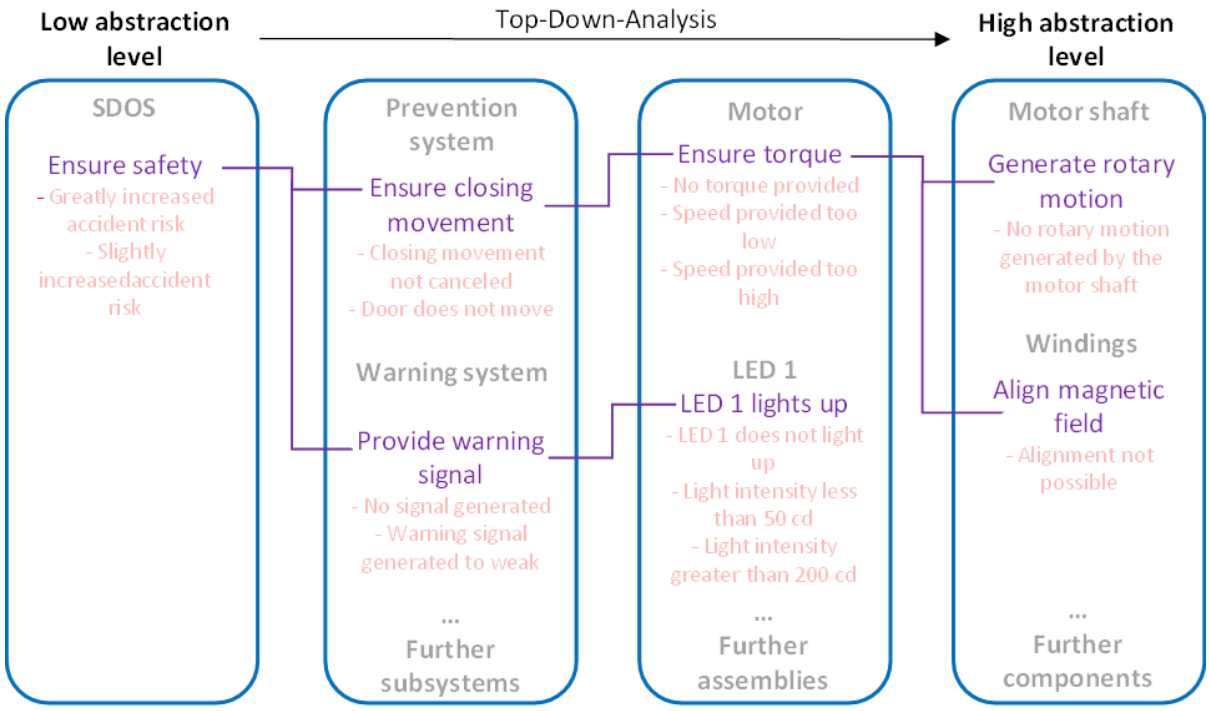

Figure 7. Functional Analysis

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

6. Summary and Outlook

Following steps include performing the same analysis with Synedia's capability of centralizing all the connections in a Cloud. This could bring new findings for a centralized tool that manages all connections for the system engineering process. The same analysis can also be performed with different tools than the ones proposed here, using for example open source solutions to integrate the system engineering process to the safety analysis. Further research shall investigate, how concurrent engineering is performed applying the proposed analysis with several engineers in parallel. Like this the impact on the change management process can be conducted plus the consistency of the trace links.

There is potential to support the trace-link establishment by automating it during the process. This might improve the acceptance of the tool, and bring the benefits of the traceability at a lower time-cost for the engineers. The main barrier is the imperfection of the software in this work. There are always some implementation problems and often all links must be established manually, which increases the time for developing models by a factor of several. Secondly, there are always problems of understanding the relationship between functions and requirements and system elements. Basically, this problem can be caused by the fact that during development, engineers are guided by the practical implementation of the product, and not by the description of its functions.

In this paper, the authors examined all aspects of modelling and established direct links between the models. In particular, it is important to note the importance of linking functions with system decisions that influence behavior patterns. Thus, it is possible to trace the connections of the direct functions of the product and implement them through simulation and tests. In this case, testing functions, through behavioral and systemic models, allows engineers to identify the nature of possible violations and problems in the product. When a new problem in product is found, it will be clear which product functions it affects and at what level of abstraction of functions it is. Thus, such an analysis determines the degree of possible threats and determines the area of problems in the implementation of product functions and solution elements.

Either way, there are the following steps to investigate traceability issues: Improved software packages to quickly establish links between objects; Description of the influence of errors of functions and system elements on the quality of the product.

References

Albers, A. and Lohmeyer, Q. (2012), "Advanced Systems Engineering - Towards a Model-Based and Human-Centered Methodology", in Tools and methods of competitive engineering: Proceedings of the Ninth International Symposium on Tools and Methods of Competitive Engineering, TMCE 2012, May 7 - 11, 2012, Karlsruhe, Germany ; digital proceedings, Faculty of Industrial Design Engineering Delft University of Technology, Delft, pp. 407–416.

Arnold, A., Point, G., Griffault, A. and Rauzy, A. (1999), "The AltaRica Formalism for Describing Concurrent Systems", *Fundamenta Informaticae*, Vol. 40 No. 2,3, pp. 109–124. 10.3233/FI-1999-402302.

Bender, K. (Ed.) (2005), *Embedded Systems - qualitätsorientierte Entwicklung*, *SpringerLink Bücher*, Springer-Verlag, Berlin/Heidelberg. 10.1007/b138984.

Biggs, G., Juknevicius, T., Armonas, A. and Post, K. (2018), "Integrating Safety and Reliability Analysis into MBSE: overview of the new proposed OMG standard", *INCOSE International Symposium*, Vol. 28 No. 1, pp. 1322–1336. 10.1002/j.2334-5837.2018.00551.x.

Bucholz, C., Tiemann, M. and Stark, R. (2018), "Durchgängiges Prototyping mechatronischer Systeme im MBSE Entwicklungsprozess", in Krause, D., Paetzold, K. and Wartzack, S. (Eds.), *Design for X: Beiträge zum 29. DfX-Symposium September* 2018, Institut für Technische Produktentwicklung, Universität der Bundeswehr München, Neubiberg, pp. 119–130.

David, P., Idasiak, V. and Kratz, F. (2010), "Reliability study of complex physical systems using SysML", *Reliability Engineering & System Safety*, Vol. 95 No. 4, pp. 431–450. 10.1016/j.ress.2009.11.015.

Estefan, J. (2008), "Survey of Model-Based Systems Engineering (MBSE) Methodologies", *INCOSE MBSE Focus Group*, Vol. 25.

Helle, P. (2012), "Automatic SysML-based safety analysis", in Ober, I. (Ed.), Proceedings of the 5th International Workshop on Model Based Architecting and Construction of Embedded Systems - ACES-MB '12, 9/30/2012 - 9/30/2012, Innsbruck, Austria, ACM Press, New York, New York, USA, pp. 19–24.

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press

INCOSE (2015), INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th ed., Wiley, New York.
ISO (2018), ISO 26262-1:2018: Road vehicles - Functional safety - Part 1: Vocabulary.
Mhenni, F., Choley, J.-Y., Nguyen, N. and Frazza, C. (2016), *Flight Control System Modeling with SysML to Support Validation, Qualification and Certification*, Vol. 49. 10.1016/j.ifacol.2016.07.076.
Riel, A., Kreiner, C., Messnarz, R. and Much, A. (2018), "An architectural approach to the integration of safety and security requirements in smart products and systems design", *CIRP Annals*, Vol. 67 No. 1, pp. 173–176. 10.1016/j.cirp.2018.04.022.
Sillitto, H., Martin, J., McKinney, D., Griego, R., Dori, D., Krob, D., Godfrey, P., Arnold, E. and Jackson, S. (2019), *Systems engineering and system definitions*.
Verein Deutscher Ingenieure (2019), VDI 2803: Functional analysis - Fundamentals and method No. 2803.

https://doi.org/10.1017/pds.2022.203 Published online by Cambridge University Press