USING MODEL-BASED SYSTEMS ENGINEERING FOR NEED-BASED AND CONSISTENT SUPPORT OF THE DESIGN PROCESS

Husung, Stephan (1);
Weber, Christian (2);
Mahboob, Atif (2);
Kleiner, Sven (1)

1: :em engineering methods AG;
2: Technische Universität Ilmenau

ABSTRACT

Model-Based Systems Engineering (MBSE) is an efficient approach to support product development in order to meet today's challenges. The MBSE approach includes methods and, above all, modelling approaches of the technical system with the aim of continuous use in development. The objective of this paper is to use the potential of the MBSE models and to show the added value of such models on the system level when used as a single source. With this objective, this paper presents a three-step approach to systematically identify and apply meaningful modelling approaches within MBSE, based on the needs during the development process. Furthermore, an FMEA example is included in this paper to elaborate the use of MBSE in the system failure analysis.

Keywords: Model-Based Systems Engineering, Systems Engineering (SE), Design methods, Product modelling / models

Contact:
Husung, Stephan
Technische Universität Ilmenau
Product and Systems Engineering Group
Germany
Stephan.Husung@tu-ilmenau.de

Cite this article: Husung, S., Weber, C., Mahboob, A., Kleiner, S. (2021) 'Using Model-Based Systems Engineering for Need-Based and Consistent Support of the Design Process', in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.598

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

1 INTRODUCTION

Engineers are facing increasing challenges in today's product development. Main drivers for these challenges are:

- higher system complexity based on multidisciplinary distributed functions, e.g. mechatronic or cybertronic systems (Eigner et al., 2017) or system of systems (SoS) containing systems that also go beyond the purely technical area - e.g. socio-technical or ecological systems,
- increased focus on the realisation of required functions with necessary reliability, sensitivity, reproducibility and availability instead of focus on components, e.g. customers want to have solutions based on functions and not necessarily based on components of a system (service-orientation instead of product focused),
- increasingly strict safety and security requirements of systems and/or legal obligations, e.g. ISO 26262, ISO 21448 in the automotive sector or FDA requirements in the medical environment,
- nationwide or globally distributed interdisciplinary development teams,
- agile development processes (Goevert and Lindemann, 2018) and systems that require further development and updating based on feedback after delivery, as well as
- the wish to capture dynamic feedback about information from the system's later life stages using digital twins in order to use it for the extraction of information, maintenance prediction or as a basis for new business models (Moyne et al., 2020).

These challenges lead to re-thinking and re-designing the development process. An efficient approach to support the handling of the system and project complexity is Systems Engineering (SE). SE is a well-known approach that is described in the Systems Engineering Handbook of INCOSE (International Council on Systems Engineering) (Walden et al., 2015) and ISO 15288. SE includes processes (not only technical) with development activities and methods (Haberfellner et al., 2019). In order to meet the challenges described above, in addition to the SE processes with development activities and methods, models are useful that consistently represent the information that has been defined or obtained in the development process (based on synthesis decisions and analysis results); this information can be used to enhance the process and to improve the traceability within the information flow (Kleiner et al., 2017). The enhancement of SE with models of the product and its functions is called Model-Based Systems Engineering (MBSE). The main focus of the models in the context of MBSE is on the mechatronic system level.

In research, numerous MBSE modelling approaches are already discussed and applied efficiently in industry today (Hoffmann, 2011; Pohl, 2012; Weilkiens et al., 2016). These modelling approaches address important development activities like:

- requirements elicitation,
- functional decomposition,
- architecture design with system decomposition,
- interface specification,
- architecture decision-making as well as
- the partial determination of verification cases.

By using MBSE approaches, frequent process instructions (among others required by ISO 26262, ISO 21448 or ISO 15504) are satisfied (e.g. decomposition of requirements, connecting requirements with architectural specifications). However, the modelling approaches do not yet cover all the development activities and methods that are required to meet the challenges of product development described above, and also they do not yet fully exploit the potential of existing MBSE methods and languages. This often leads to the situation that the models are built in parallel to the existing development process as a secondary source of information and not as a single source (as information core). This raises questions on the benefit/effort ratio, the consistency and validity of the information saved in MBSE models.

The objectives of this paper are to utilise the potential of MBSE models to address the challenges in today's product development and to show the added value of such models on the system level when they are used as a single source (or at least federated single source for distributed models - often with different levels of detail and formalisation). With these objectives, this paper presents a three-step approach that starts with the systematic identification of modelling needs based on the activities during the development process, followed by concrete modelling proposals and ends on the consolidation of the necessary model elements and views. Furthermore, the development of this approach is based on

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

the study and analysis of well-known SE methods and their implementation using the most frequently applied MBSE language, i.e. Systems Modeling Language (SysML).

On the whole, two research questions are addressed in this paper:

- Which development activities can be supported during the system development process by MBSE methods and specific modelling approaches to get closer to using MBSE models as single source?
- Which model elements and views of the model elements can support the development activities in the context of specific MBSE methods?

2 STATE OF THE ART

SE is a holistic and robust approach for efficient and successful development of complex technical systems and related services by the systematic use of **processes** and **activities** (Walden et al., 2015). The technical processes of SE consist of atomic **development activities** (specific steps within the processes) and **methods** for the support of the development activities (Haberfellner et al., 2019), namely for requirement elicitation, specification and refinement, functional, logical and technical system specification and design, system integration, verification and validation. In addition, management activities, such as for configuration and change management, are defined.

MBSE is an extension of SE and, in contrast to traditional document-based engineering, works with semi-formal models to represent information about the product to be developed. The model as a digital representation of the product is created during the development activities based on design decisions and analysis results. It will be made available for follow-up development activities. The system representation in MBSE is called "**system model**". The most common modelling language for system models is SysML (Friedenthal, 2015; OMG, 2019). With this language, the system model is described as a hierarchical decomposition of so-called "**system elements**" (representation of the mechatronic, mechanical, electrical/electronic, software parts of the product). The system elements are modelled with interfaces for the interactions between them or with the outside world. Each system element can be further detailed in terms of description of function, behaviour and other properties. For the creation of the **model elements** (atomic elements for the description of the system elements, behaviour, properties, interactions, etc.) as well as for further use, **views** on the system model are used, like filters that extract model elements that are relevant for specific contexts and tasks. The views include structural, behavioural and requirement related model elements.

Several MBSE modelling approaches are currently available for the concrete description of modelling in the context of specific MBSE activities. Table 1 shows significant and relevant MBSE modelling approaches available and frequently applied in industry today, along with a mention of the development activities that could be supported by them (alphabetical order).

Table 1: MBSE modelling approaches

MBSE modelling approaches	**Main addressed development activities during the design**
Functional Architecture for Systems (FAS) (Weilkiens et al., 2016)	Elicitation of requirements based on Use Cases, functional decomposition, grouping of the functions
Harmony for Systems Engineering (SE) (Hoffmann, 2011)	Elicitation of requirements, functional decomposition (static and dynamic), logical architecture design and interface specification, architecture decisions
Object-Oriented Systems Engineering Method (OOSEM) (Friedenthal, 2015)	Elicitation of requirements, functional decomposition (static and dynamic) on different system levels, logical decomposition and architecture design and interface specification, optimisation and evaluation alternatives, system validation and verification
Software Platform Embedded Systems (SPES) Method (Pohl, 2012)	Elicitation of requirements, functional decomposition (static and dynamic) on different system levels, logical decomposition and architecture design and interface specification, architecture decisions
Systems Modeling Toolbox (SYSMOD) (Weilkiens, 2016a)	Elicitation of requirements, functional decomposition, logical and physical architecture definition

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

There are also some more approaches available and a few of them are mentioned in (Eigner et al., 2017). All modelling approaches mainly cover the left branch of the V-model concept for development processes (VDI, 2004). There are individual approaches for expansion, e.g. regarding the mapping of variants (Akin et al., 2019; Weilkiens, 2016b). However, the information represented by MBSE models (e.g. concerning requirements, functional and logical architecture) offers even greater potential for use in product development, since the information in the models is needed for many other activities (e.g. functional and logical architecture for carrying out system analyses or as a basis for system verification). Based on the relevant activities during the development process and the necessary methods and information, the approach presented in this paper aims to show how the MBSE models can be extended in a goal-oriented way and used for further development activities.

3 APPROACH

3.1 General Approach

The following three **steps** are proposed by the approach in order to identify the relevant development activities in the development process, that can meaningfully be supported by MBSE modelling and the related modelling approaches:

1. Analysis of the development activities (assumption: the development activities are already known) during the development process regarding the necessary inputs and outputs (work products) in relation to the methods used. The question here is: Which development activities require information as input, process information or generate information as output that can be well represented using semi-formal MBSE modelling.
2. Detailed analysis of the individual methods during the relevant development activities: for example, there exist well-known questioning methods (e.g. according to VDI 2222); for risk and reliability analysis, there are well-known methods like FMEA and FTA available; for verification and validation, ISQTB and other standards define many test design techniques (like equivalence class analysis). For each method:
    - The elementary information artefacts and their interrelation are determined. Furthermore, it is determined which views on the information artefacts are necessary/sensical for the representation and usage (e.g. analysis) of the information.
    - Analysis of the mapping of information artefacts and their interrelation using the MBSE description language (see also figure 1).
    - Derivation of the modelling steps necessary to create the MBSE artefacts and their usage.
3. Consolidation of all model elements and views required for the relevant development activities and methods, including their modelling steps, into a holistic modelling approach.

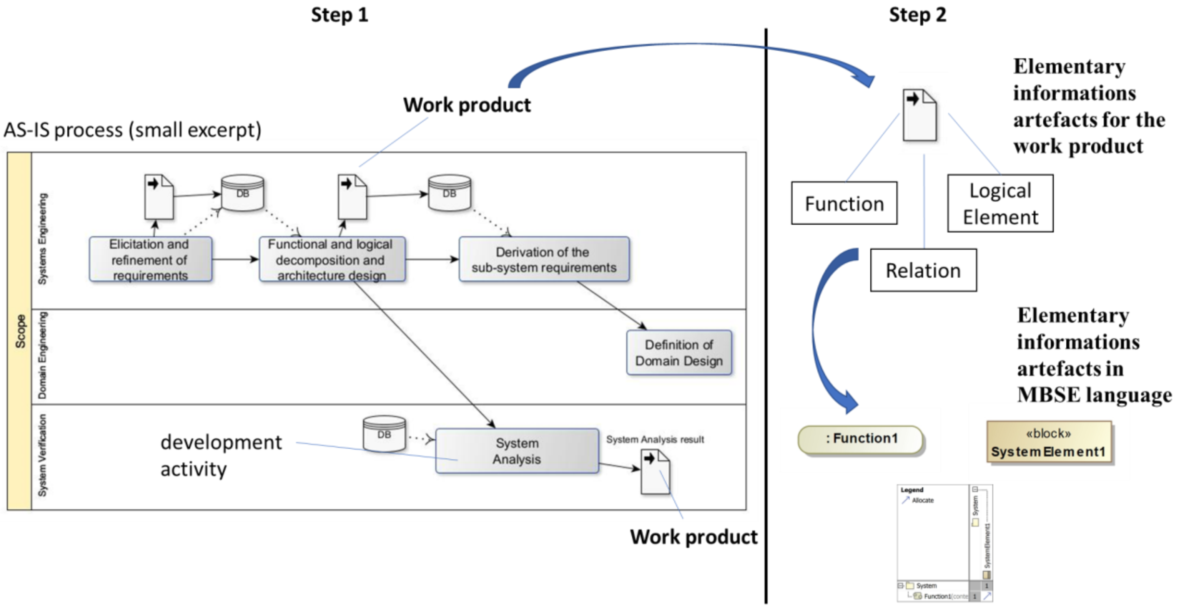

Step 1
Step 2
Work product
AS-IS process (small excerpt)
Scope
Systems Engineering
Domain Engineering
System Verification
DB
Elicitation and refinement of requirements
Functional and logical decomposition and architecture design
Derivation of the sub-system requirements
Definition of Domain Design
development activity
System Analysis
System Analysis result
Work product
Elementary informations artefacts for the work product
Function
Logical Element
Relation
Elementary informations artefacts in MBSE language
: Function1
«block»
SystemElement1
Legend
Allocate
System
SystemElement1
Function1(conte
1

Figure 1: Step 1 and 2 of the approach

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

Based on the experience of the author group in several industrial projects, a typical set of development activities that can be supported by MBSE modelling approaches are visualised in figure 2, i.e. requirements elicitation, requirements and system decomposition, derivation of requirements of the subsystems, analyses, etc.

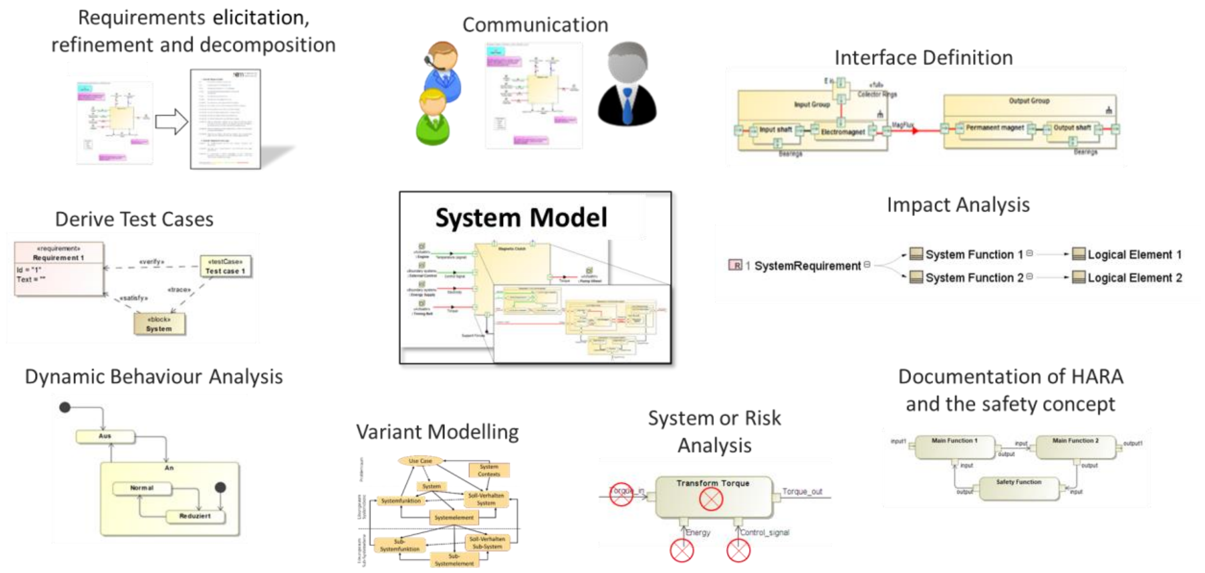


Figure 2: Typical high-level MBSE supported development activities, supplements from (Husung et al., 2018)

Further typical development activities are e.g. documentation of decisions, configuration management, representation of knowledge (e.g. malfunctions and their causes), tracking of system maturity or definition of a basis for digital twin models (see table 2). Often development teams start with MBSE support of “simple” development activities like interdisciplinary communication or requirement elicitation and decomposition (Husung and Kleiner, 2018; Kleiner et al., 2017). For these MBSE modelling approaches, primarily informal views on the system model elements can be used. Important for these development activities and MBSE modelling approaches is a common understanding that can be promoted by populating the views collaboratively. Table 2 addresses the second research question from section 1 and lists possible development activities that can be supported with MBSE as well as their descriptions and the added value brought by using MBSE. The entries in table 2 are again arranged in alphabetical order. Furthermore, a number of development activities are included in table 2 that directly address the challenges in today's product development (see section 1).

Table 2: Typical high-level MBSE supported development activities

MBSE supported development activity	**Description**	**Added Value**
Burden of proof, certification	Documentation for obligatory verification (e.g. ASPICE, FuSi)	Reduction of effort by use of a common architecture, documentation and proof of evidence; reduction of errors by direct integration
Definition and tracking of maturity levels	Defining maturity levels for system, functions and system elements. Tracking of maturity level during the development (e.g. implementation of requirements)	Improvement of the traceability; identification of implementation progress and coverage
Definition of a base for digital twin models	Based on the use cases of the digital twin, the necessary information model with the associated behaviour and parameters (the base for the digital shadow) is defined	Consistent and comprehensible description, reduction of errors, creation of a base for adjustments during later life phases

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

Elicitation and decomposition of requirements	Systematic decomposition of requirements according to life phases, use cases, scenarios as well as the defined functional and logical architecture (concept)	Development process strictly based on requirements; development focus on important product aspects
Failure analysis	Systematic identification and traceability of failures in the system (or system description) by applying defined analysis methods	Improvement of the system quality; prevention and reduction of errors
Generation of domain-specific architectures	Generation of domain-specific architectures based on the logical or physical system architecture	Reduction of errors (single and traceable source)
Impact analysis (e.g. for change management)	Systematic performing of impact analyses based on change requests on different system levels	Improvement of the quality of impact analyses; time reduction for impact analyses; shorter change and iteration cycles
Interface specification and coordination	Systematic and holistic description of interfaces, flows and parameters between subsystems	Consistency between interfaces; efficient interface analysis and documentation; reduction of coordination effort; reduction of errors during coordination between developers
Know-how capturing	Creating libraries of building blocks and knowledge (e.g. concepts, decisions, allocated methods, systems, solutions, etc.). Representation of lessons learned.	Reduction of effort by model-based knowledge representation, transparent expert and experience know-how, better knowledge transfer
Parameter management	Representation of the necessary requirement (required parameters), design (defined parameters) and test (as-is parameters) parameters in the system model	Reduction of effort based on a uniform database and systematic re-use, more efficient configuration and variant calculation
Planning of system integration	Systematic planning of the system integration (integration strategy)	Comparison of system architecture alternatives; clear context and interface definitions for system elements; reduction of coordination effort; reduction of errors and incompatibilities (single and traceable source)
Rational decision-making and documentation	Create a base for systematic decisions and document the decisions for further process steps	Transparent and traceable decision rationale; reduction of errors during the decision process; reduction of coordination effort
System analyses for safety, security and reliability	Use of the already defined functional, logical or physical architecture and the allocation between the model elements in the different tools for safety, security and reliability	Reduction of effort and errors by continuous use of the system architecture
System of Systems (SoS) analyses	Analysis of the use cases and the interaction of the systems involved (technical but also socio-technical, ecological, etc.). A major challenge here is to bring together the different life cycles of the individual systems within the SoS.	Performing holistic analyses, consistent and comprehensible description

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

System understanding, communication and collaboration (horizontal and vertical, incl. external partners)	Create a common understanding of all relevant interdisciplinary development partners based on a consistent database	Clearer system relations and interactions; reduction of misunderstandings, reduction of coordination effort
Variant representation and instance generation	Representation of a 150% system model with the requirements, functional and logical/physical system elements for system configuration. Analysis of different variants. The base for 100% solutions.	Consistent and manageable relationships between system elements and variants; Reduction of variation complexity; Reduction of errors (especially at the interfaces of the elements)
Verification and validation (V&V)	Systematic definition of verification/ validation criteria and measures, test cases e.g. for integrating tests and applying test design techniques like state analysis	Traceability between requirements, criteria and measures; better understanding of V&V coverage; improvement of test case definitions (quality and completeness)

Each of the development activities mentioned in table 2 could be explained in further details and the modelling process could be explained to show the added value. As an example, the system analysis activities from table 2 are explained in more detail to show how MBSE modelling can support them. The focus of the explanations is on step 2 (see figure 1), the determination of the necessary modelling elements and views is based on the methods used during the development activities.

3.2 System Analyses

A few of the relevant development activities during the development process are system analysis activities (many others are of course also important). Among other things, these analysis activities are relevant to assure product safety, functional safety or reliability for the appropriate stakeholder. A well-known method for safety and reliability is the Failure Modes and Effects Analysis (FMEA). In order to perform an FMEA, representations of function structure and logical or physical system element structures are necessary. In a conventional approach, these models do not exist in the first place and have to be determined in hindsight in an FMEA workshop, i.e. late in the development process. Using MBSE, the preparation of the FMEA can be based on an already existing function structure representation (in MBSE language: functional architecture and activities), the logical and/or physical architecture and the mapping between these two in the system model. Using the system model elements and views for the functions, potential malfunctions, causes and effects can be identified systematically (see figure 3).

The main methodological steps for the FMEA are given inside table 3 in column 2.

For structural analysis, the decomposition of the system and, at each system level, the relations between the system elements are required. This information is implemented in the MBSE modelling by means of structural elements, i.e. "Blocks", their decomposition, and for the relations between them "Ports" and "Connectors" are used.

For the functional analysis, the functions relevant to the system level and their relationships along the functional effect chain or network are required. This information can be derived from the functional architecture, which is a result of system development. In the MBSE modelling language SysML, the information artefacts correspond to "Activities", which are related via "Pins" and "Object Flows" as well as other relationships.

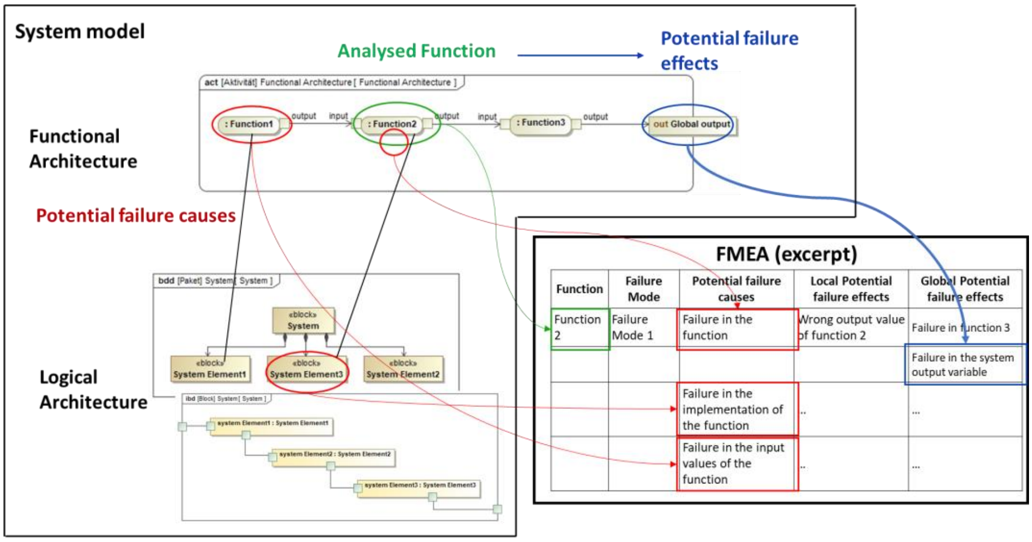

System model
Analysed Function
Potential failure effects
act [Aktivität] Functional Architecture [Functional Architecture]
: Function1
output
input
: Function2
output
input
: Function3
output
out Global output
Functional Architecture
Potential failure causes
bdd [Paket] System[System]
«block» System
«block» System Element1
«block» System Element3
«block» System Element2
Logical Architecture
ibd [Block] System[System]
system Element1 : System Element1
system Element2 : System Element2
system Element3 : System Element3
FMEA (excerpt)
Function
Failure Mode
Potential failure causes
Local Potential failure effects
Global Potential failure effects
Function 2
Failure Mode 1
Failure in the function
Wrong output value of function 2
Failure in function 3
Failure in the system output variable
Failure in the implementation of the function
..
...
Failure in the input values of the function
..
...

Figure 3: Preparation of the FMEA based on the function and logical architecture in the system model

For failure modes analysis, possible malfunctions of the individual functions are required. Along the functional chain/network of actions, the possible causes and effects of faults (local and global) can be determined. The analysis of the failure causes can be carried out over several functional decomposition levels and up to the analysis of the system elements which implement these functions. For this purpose, the allocations between the functions and system elements are evaluated. In table 3, the relevant SysML model elements needed for the implementation of the given methodological steps are listed.

Table 3: SysML Model elements and diagram types (names see (OMG, 2019)) for FMEA working steps

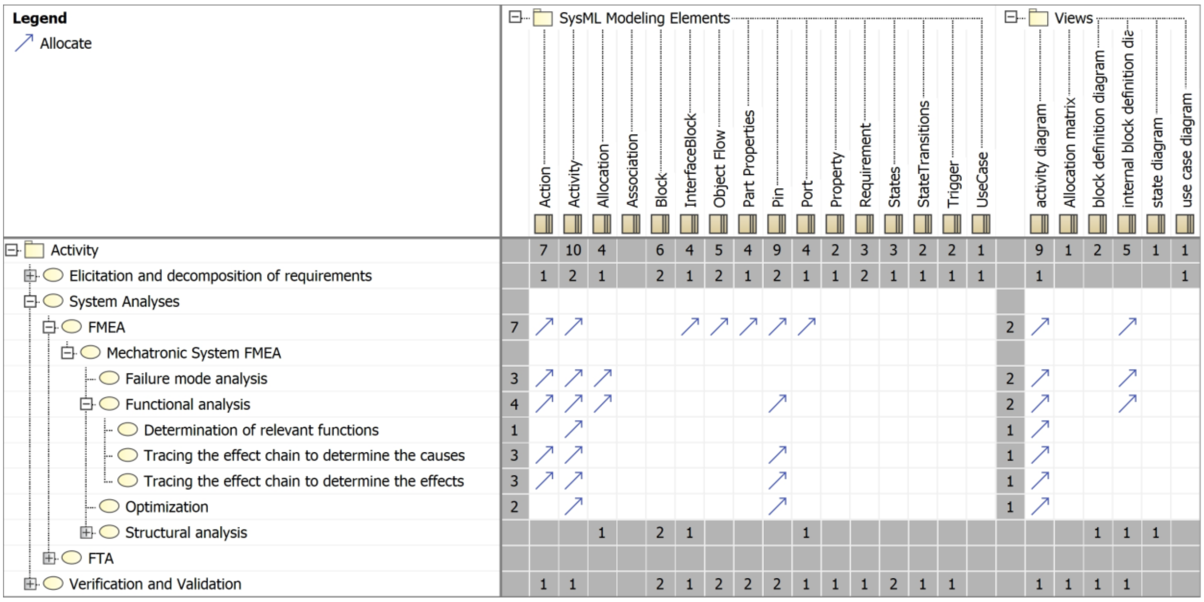
Nr.	Methodological step	Description	SysML diagram type	SysML model elements
1.	Structure analysis	Identification of system elements and relations	Structure diagrams	Blocks, Part properties, Ports, Connectors
2.	Function analysis	Identification of system functions and their effects	Behaviour diagram	Action, Activity, Control flow, Object flow, Fork/Join and Decision/Merge elements
3.	Failure mode analysis	Identification of failure modes for the necessary system functions	Behaviour and structure diagram	Action, Activity, Control flow, Object flow, Fork/Join and Decision/Merge elements, State, Transition, Part properties, Ports, Connectors
4.	Definition of preventive measures	Documentation and assessment of detection and avoidance measures	Not part of the approach	
5.	Optimisation	Reduction of failure modes and (assessment of the optimised solution - not a part of the approach)	Behaviour and structure diagram	Action, Activity, Control flow, Object flow, Fork/Join and Decision/Merge elements, State, Transition, Part properties, Ports, Connectors

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

The use of the system model created during system development including the functional and logical architecture reduces the effort (e.g. for dedicated workshops) and allows the FMEA to grow along the system development, making it a more efficient method for system analysis.

3.3 Consolidation of development activities and definition of the necessary modelling scope

For individual development activities, the approach results in an overview of relevant modelling elements and views, as shown by an example in section 3.2. Step 3 of the approach focuses towards the collection and consolidation of the identified modelling elements and views across all development activities. The consolidation of these results in an overall representation of the system and its behaviour, which, if modelled using MBSE, can support the relevant development activities. Figure 4 shows a possible form of a consolidated overall model represented as an allocation matrix. In the top row, the number of uses of the modelling elements and views across the development activities can be seen.



Legend
↗ Allocate

	SysML Modeling Elements	Action	Activity	Allocation	Association	Block	InterfaceBlock	Object Flow	Part Properties	Pin	Port	Property	Requirement	States	StateTransitions	Trigger	UseCase	Views	activity diagram	Allocation matrix	block definition diagram	internal block definition die	state diagram	use case diagram
Activity		7	10	4		6	4	5	4	9	4	2	3	3	2	2	1		9	1	2	5	1	1
Elicitation and decomposition of requirements		1	2	1		2	1	2	1	2	1	1	2	1	1	1	1		1					1
System Analyses																								
FMEA	7	↗	↗				↗	↗	↗	↗	↗							2	↗			↗		
Mechatronic System FMEA																								
Failure mode analysis	3	↗	↗	↗														2	↗			↗		
Functional analysis	4	↗	↗	↗						↗								2	↗			↗		
Determination of relevant functions	1		↗															1	↗					
Tracing the effect chain to determine the causes	3	↗	↗							↗								1	↗					
Tracing the effect chain to determine the effects	3	↗	↗							↗								1	↗					
Optimization	2		↗							↗								1	↗					
Structural analysis				1		2	1				1										1	1	1	
FTA																								
Verification and Validation		1	1			2	1	2	2	2	1	1	1	2	1	1			1	1	1	1		

Figure 4: Consolidation of the SysML model elements and views for the relevant development activities

The system has different system levels for which different development activities and thus different MBSE modelling approaches can be useful. For larger systems, therefore, a more fine-grained determination of the MBSE model elements for each system level is useful.
(Albers et al., 2015) show that in most practical development tasks large parts of a predecessor system, at least on the sub-system level, are reused (Product Generation Development). Therefore, it is usually not necessary to conduct system modelling activities from the beginning every time. Instead, especially functional and logical chains/networks of actions, states or interface descriptions can be reused, at least on the sub-system level. Such a reuse is also an added value when using MBSE for the support of development activities.

4 SUMMARY AND OUTLOOK

The approach described here demonstrates that, based on an analysis of the activities in the development process, the most meaningful MBSE model elements and views that support these development activities can be identified. This helps avoiding redundant information definitions during development. The MBSE model can be used as a single source for carrying out multiple activities. Also, the MBSE models are only built in such a way that they contain the minimum information about artefacts, to only what is necessary for the follow-up development activities. This allows keeping the focus and concentration during the modelling activities.
The paper addresses the content of the research questions mentioned in the introduction to identify the development activities to be supported and determine the scope as well as detail of modelling. Depending on the implementation of models and further methods in integrative or federal tool environments,

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press

further research questions may arise regarding data management (especially regarding configuration management) as well as data exchange.

In future research work, the modelling approaches for additional development activities will be further detailed and evaluated in concrete development projects. In addition, concepts for the efficient use of the approach will be further developed, including usability as well as data management and data exchange aspects.

REFERENCES

Akin, M., Grimm, V., Husung, S., Stöckeler, C., Hahn, D. and Kleiner, S. (2019) 'Beherrschung komplexer mechatronischer Systemvarianten mittels MBSE und SysML', *Tag des Systems Engineering*, 6. - 8. November 2019. München.

Albers A., Bursac N. and Wintergerst E. (2015) 'Product Generation Development – Importance and Challenges from a Design Research Perspective'.

Eigner, M., Koch, W. and Muggeo, C. (2017) *Modellbasierter Entwicklungsprozess cybertronischer Systeme*, Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-55124-0

Friedenthal, S. (2015) *A practical guide to SysML: The systems modeling language* [Online], Waltham, MA, Morgan Kaufman. Available at http://proquest.tech.safaribooksonline.de/9780128002025.

Goevert, K. and Lindemann, U. (2018) 'Further development of an agile technique toolbox for mechatronic product development', *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 2015–2026. https://doi.org/10.21278/idc.2018.0204

Haberfellner, R., Weck, O. L. de and Fricke, E. (2019) *Systems engineering: Fundamentals and applications*, Springer International Publishing. https://doi.org/10.1007/978-3-030-13431-0

Hoffmann, H.-P. (2011) *Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering - Deskbook Release 4.1: Model-Based Systems Engineering with Rational Rhapsody and Rational Harmony for Systems Engineering*.

Husung, S. and Kleiner, S. (2018) 'MBSE and Simulation for Automotive Systems Engineering', *NAFEMS Seminar: Bessere Produkte mit Modellbasiertem Systems Engineering (MBSE) und CAE.* Wiesbaden.

Husung, S., Lindemann, G., Korobov, S., Hamester, M. and Kleiner, S. (2018) 'Use Case driven Model-based Systems Engineering for industrial applications', *EMEASEC 2018 / TdSE* 2018.

Kleiner, S., Ricks, M., Engelmann, M. and Husung, S. (2017) 'Model-Based Systems Engineering and Simulation for Automotive Systems Development at GKN Driveline', *NAFEMS World Congress (NWC17).* Stockholm, 11.-14.06.2017.

Moyne, J., Qamsane, Y., Balta, E. C., Kovalenko, I., Faris, J., Barton, K. and Tilbury, D. M. (2020) 'A Requirements Driven Digital Twin Framework: Specification and Opportunities', *IEEE Access*, vol. 8, pp. 107781–107801. https://doi.org/10.1109/ACCESS.2020.3000437

Object Management Group (2020) *UAF Overview* [Online]. Available at https://www.omgwiki.org/uaf/doku.php.

OMG (2019) *The OMG Systems Modeling Language™ Version 1.6* [Online].

Pohl, K. (2012) *Model-based engineering of embedded systems: The SPES 2020 methodology*, Heidelberg, Springer. https://doi.org/10.1007/978-3-642-34614-9

VDI (2004) 2206*:2004-06: Design methodology for mechatronic systems*.

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D. and Shortell, T. M., eds. (2015) *Systems engineering handbook: A guide for system life cycle processes and activities ; INCOSE-TP-2003-002-04*, 2015, 4th edn, Hoboken, NJ, Wiley.

Weilkiens, T. (2016a) *SYSMOD - the systems modeling toolbox: Pragmatic MBSE with SysML*, 2nd edn, Fredesdorf, MBSE4U.

Weilkiens, T. (2016b) *Variant modeling with SysML*, Fredesdorf, MBSE4U.

Weilkiens, T., Lamm, J. G., Roth, S. and Walker, M. (2016) *Model-based system architecture*, Hoboken, NJ, Wiley.

https://doi.org/10.1017/pds.2021.598 Published online by Cambridge University Press