Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML

Albert Albers and Christian Zingel

IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT), Kaiserstr. 10, 76131 Karlsruhe, Germany
{albert.albers,christian.zingel}@kit.edu

Abstract. Model-Based Systems Engineering is on everyone's lips as innovative approach to overcome traditional, error-prone document-based product development. The Systems Modeling Language (SysML) is the most popular tool for model-based development of multidisciplinary systems. Several research works and industrial pilot projects have applied the OMG-standardized language in the last years, but it has still not become widely accepted. Previous experiences of the authors from several research projects with industry underline this statement and have shown that engineers still have trouble in applying SysML. This paper investigates possible reasons for this issue and presents results of a survey regarding term understanding of engineers as well as acceptance of SysML.

Keywords: Term understanding, Common Engineering Language, SysML, Systems Modeling Language, Acceptance.

1 Model-Based Systems Engineering – Potentials and Challenges

Model-Based Systems Engineering (MBSE) is on everyone's lips as an innovative approach to overcome traditional, error-prone, document-based product development. The advantages of using formal models to specify a complex technical system are manifold: fewer inconsistencies, less redundancies and concurrently the basis for clear communication and sustainable documentation. Furthermore, models can help to force systemic thinking, which is often expressed as holistic and function-based thinking [1]. Over the last years, several modeling languages and approaches have been presented. They all promised to enable their users to model multidisciplinary and complex technical systems. The most popular modeling language is SysML, which is based on UML, a widely accepted, object-oriented graphical software modeling language. Observations of the authors have shown that software engineers and electronics engineers cope well with the provided diagrams for modeling several system aspects like structures, sequences or states. Emerging graphical modeling languages for embedded systems like MARTE, AUTOSAR or EAST-ADL using similar

M. Abramovici and R. Stark (Eds.): *Smart Product Engineering*, LNPE, pp. 83–92.
DOI: 10.1007/978-3-642-30817-8_9

principles and technologies like the meta-modeling standard MOF (Meta Object Facility) seem to underline this observation. On the contrary, mechanical engineers have much more trouble dealing with such kind of modeling languages. A possible reason is that technical terminologies between disciplines differ significantly. Furthermore, there is a huge leap of abstraction from concrete discipline-specific models to abstract multi-disciplinary system models. An online survey presented in this paper was conducted in order to identify the understanding of some crucial terms in the field of MBSE and to determine the state of application of SysML. The survey was sent to selected engineers from different disciplines in industry and academia in Germany, who are familiar with Systems Engineering (i.e. members of the German Chapter of INCOSE, the GfSE). Knowing that the answers will not be representative for all engineers, it was rather intended to provide an indication about term understanding and SysML application among "Systems Engineers" as spearheads in establishing MBSE in academia and industry. 50 responses (23 from academia, 27 from industry) were evaluated. Before presenting findings from this survey, the next chapter will give an overview of adjacent research efforts and their results.

2 State of Research

Several studies and academic or industrial pilot projects aimed to gain insights about the applicability of MBSE methods and tools. The ProSTEP iViP society conducted a survey in cooperation with the Fraunhofer IPK called "PEP2015 – Challenges in modern Product Engineering Processes", which evaluated needs and visions of industry and tool vendors in terms of Systems Engineering [2]. The results showed that Systems Engineering methods are applied only occasionally within software engineering and electrics/electronics engineering. Discipline-specific tools are well-established; transdisciplinary system architecture interaction is still an unsolved issue in industrial practice. Bone and Cloutier determined from another study that especially large companies are widely aware of the benefit of MBSE and increasingly adopt corresponding programs and projects [3]. Their focus is set on architecture modeling, requirements traceability and conceptual design of products. Thus, the value for software and systems engineers is much more obvious than for hardware engineers or managers. Existing organizational structures are frequently not compatible with transdisciplinary systems engineering [1], which can be substantiated by missing methodologies for the application of existing standards or modeling languages. Therefore, Estefan conducted a survey on the most prominent MBSE methodologies in 2008, aiming to mainstream them in industrial application [4]. None of those methodologies has significantly established over the last years after this survey. Kasser discusses seven myths of Systems Engineering, due to persistent discussions about possible reasons for the lacking acceptance and application in industrial product engineering [5]. He found out, that there is neither a single broad agreement upon systems engineering processes nor on the adequate application of tools and methods to handle system complexity.

Several pilot projects have taken place in order to determine best practices or to evaluate first applications of MBSE tools and methods. Friedenthal presented several findings from the application of SysML [6]. He stated that MBSE is a cultural change and requires well-defined methodologies and handling them requires training in language, methods and tools. Karban et al. state challenges in using SysML, which have been figured out in the APE (Active Phasing Experiment) project of the SE^2 challenge team of the GfSE [7]. They propose several tasks for the advancement of SysML, which underlines that the language is still under development and will be further advanced in the future.

Other research efforts deal with the definition and understanding of frequently used terms in engineering disciplines. An example for such a term is “function”, which is for instance understood in software engineering as a piece of software code that processes input information towards a certain output. Mechanical engineers on the contrary have different connotations for “function”: it can either describe, what a system to develop is intended to do or what a system solution actually does. Moreover, a function is often distinguished between a desired function and an undesirable one. Others would name an undesired function an appearing phenomenon, an effect or a behavior. Several efforts have addressed the understanding of terms, so has Eckert et al. [8] for instance investigated the different notions of the previously mentioned example “function” in engineering design. They identified the 5-key-concept of Vermaas as the most valuable, but also differentiating definition of this term, meeting most of the previously mentioned examples [9]. Vermaas concludes that different meanings are required in different situations instead of pursuing a single definition of “function” through emphasizing that different meanings are in fact necessary to describe devices in engineering design.

Literature review has shown that the challenges in application of MBSE are manifold and leads to the awareness, that the “cultural change” from traditional document-based towards a model-based development approach has still not taken place. The aim of this paper is to identify the cause for this issue and to point out fundamental actions to be taken in order to advance MBSE tools and methods.

3 Motivation for Further Research

MBSE aims to improve communication and collaboration between engineers from different disciplines and management. Communicating efficiently means to easily gain the desired information, provided in a comprehensible and coherent manner, which is one basic goal of MBSE. Unified term understanding is crucial for establishing a coherent, formal and coincidentally intelligible modeling language for multidisciplinary systems. Considering all relevant terms and every specialized discipline would either lead to a very generic solution like SysML or to a very extensive set of specific languages. Even if the idea to apply a common language for all involved individuals is promising, none of the existing approaches has established in industrial development yet. Possible reasons are a persistent lack of common term understanding or insufficient information representation within existing modeling languages.

The approach at hand concentrates on engineers, who are familiar with Systems Engineering paradigms or concerned with Systems Engineering research. They form the basis for the harmonization of term understanding and establishing a common language in industrial product development. The aim of the survey conducted by the authors of this paper is to answer two research questions:

- What is the understanding of the basic terms "function", "behavior" and "impact chain" among Systems Engineers?
- To what extent is SysML applied, what is the perception of the added value of SysML for the daily work today and where is improvement potential?

The results of this survey shall help to harmonize term understanding by clustering consistent statements and complement previously identified definitions. Furthermore, the demand for certain modeling aspects shall be identified and the suitability of provided diagrams in SysML for describing those aspects shall be evaluated. The long-term goal of this ongoing research work is to advance MBSE languages and coevally according modeling approaches.

4 An Approach towards a Unified Term Understanding

The presented data in this chapter result from a survey, conducted among German and Austrian Systems Engineers from academia and industry. Altogether, 50 responses (23 from academia, 27 from industry) have been evaluated. The academic participants are PhD-students (14 out of 23), students or postdoctoral researchers. The industrial participants range from development engineers over trainers and consultants to product-, project- and department-managers. The spectrum of the participants' expertise is shown in Fig. 1.

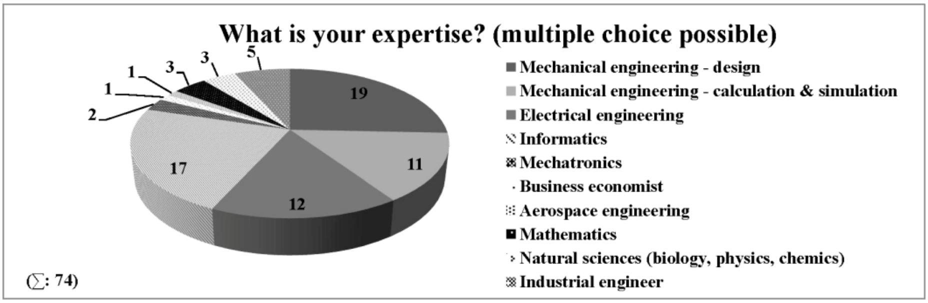

Fig. 1. Range of expertise of survey participants

The participating Systems Engineers have expertise is numerous disciplines, which helps to gain information from diverse viewpoints. However, this survey is not intended to meet representative statements with statistical evidence, but rather to point out tendencies and to collect statements from experts. The survey was divided into two sections: the first section asked for the personal understanding or definition of the three terms "function", "behavior" and "impact chain". The answers regarding term understanding were given as free statements, whereas every participant could state multiple answers per definition (i.e. to subdivide the definitions into multiple aspects). Altogether, the participants posed 78 statements for "function", 70 for "behavior" and 58 for "impact chain". The answers were characterized by a high degree of diversity. In the following, frequently made contradictory statements are contrasted:

Table 1. Contrasting pairs of statements towards the term "function"

Functions describe the purpose of a system	Functions realize functional requirements
Functions describe the role of persons in a company	Functions describe the transformation of matter/energy/information inputs into according outputs
Functions are solution-neutral	Functions are solution-afflicted
Functions are abstract specifications of transformations	Functions can be described in mathematical terms
Functions describe an active behavior	Function are an interaction of components to achieve a certain behavior

Concluding, the interpretation of the term "function" is very heterogeneous, even among Systems Engineers. The next question asked for the definition of "behavior" with explicit distinction to "function". Behavior is often associated with the performance of functions, but beyond that the survey identified a very diverse understanding of this term. Many statements differentiate between static functions and dynamic behavior, regarding the latter as a time-dependent aspect. Where functions only define the desired behavior, the description of behavior itself can also comprise misconduct (i.e. system crash) or undesired functions (i.e. noise emission). Behavior is often seen as system reaction towards environmental input stimuli under certain boundary conditions and (measureable) characteristics, which are differentiated between discrete (i.e. the event "press button") and continuous (i.e. transmit torque). Unfortunately, the opinions occasionally interfere with others and there is a lack of common understanding. For instance, one participant stated that functions are perceivable, another attributes this to behavior. Several statements contradict others regarding the question whether "behavior" describes the external view and "function" the internal view on a system or vice versa. Some participants confine behavior on the transition between system states, others acknowledge behavior to be component-afflicted; still others attest behavior to be uncontrollable. Concluding, the statements were highly diverse, but none of them embraced all mentioned aspects, which indicates a lack of unified understanding of behavior. Some statements told behavior to be a "chain of functions". Where functions are often modeled as tree structures or using logical control flows and the input-processing-output-principle for object flows, "impact chains" intend to represent a certain sequence of functions. The resulting statements regarding the understanding of "impact chain" are discussed in the next paragraph.

In contrast to "function" and "behavior", this term was not known to every participant. Two of them wrote that they had never heard this term before. The majority of the statements define an impact chain as a chain of functions, where input values of a function are the output values of the previous function. Some participants regarded these chains as high-level linking of systems, others as the internal progress of activities within a function which results in a system behavior. Some statements again contrasted: they described impact chains as synonym to traceability from requirements over functions towards implementation and test. Impact chains and active structure sound similar in German language ("Wirkketten" and "Wirkstrukturen"), but are fundamentally different. Impact chains deal with functions, active structures or working structures with components. The appearance of several terms in different meanings can lead to communication being confusing. This is why the next paragraph presents a graphical proposal (Fig. 2), illustrating semantic contexts of the important and frequently reoccurring term "function". The goal of this graphical representation is to harmonize the understanding of semantic coherences between frequently reoccurring terms in Systems Engineering. The depicted aspects embrace literature research as well as own experiences made in several development projects.

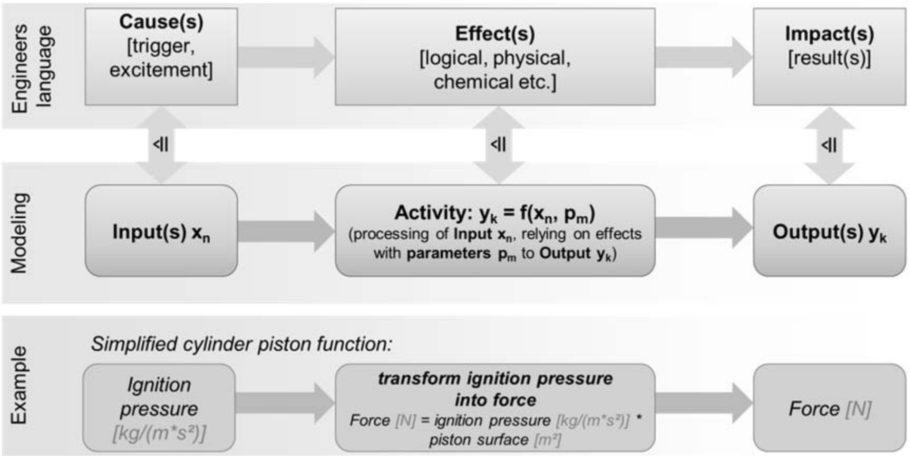

Fig. 2. Proposal for semantic context of "function"

The definition of function at hand embraces the approach to regard a function as an activity, which processes input values (information, energy, material) to output values. An input equals a cause in terms of triggering (discrete, i.e. "button pressed") or exciting (continuous, i.e. "torque flows") a function by certain input values. The processing of the input values in activities relies on logical (software), physical or chemical (mechanical and electrical systems) effects, which can be specified by equations, presuming a comprehensive knowledge about the system. The output flows have an impact on other functions (they can trigger or excite other functions) or result in perceivable phenomena (i.e. forces, noise or fields). Additionally, function-relevant physical parameters are factored into the processing of flows. Therefore, from the viewpoint of the authors, functions are not completely solution-neutral, but they are also not component-afflicted. The important awareness is to consider relevant properties for feasibility of functions, but not to anticipate an entire solution (i.e. component). This is one of the basic principles of the Contact & Channel – Approach (C&C²-A) for the integrated analysis and synthesis of functions and form of technical systems [10]. The graphical representation at hand was part of deriving and formalizing extending aspects for a SysML-profile in order to better meet the requirements of engineering designers towards application of MBSE [11], [12]. The simple example in Fig. 2 intends to make these theoretical coherences more tangible. A similar graphical representation was presented to the survey participants subsequently to the questions regarding term definitions in order not to influence or even to distort the respective statements. The participants were asked to rate the applicability of five statements concerning the presented figure. The results are depicted in Fig. 3 and discussed subsequently.

Statement in survey to rate:	**well applicable**	**partly applicable**	**not applicable**	**no rating**
A cause is an input and coincidentally the trigger of a processing activity.	68%	12%	8%	12%
An effect relies on mathematical, physical or logical principles and can be expressed in equations.	50%	38%	4%	8%
An impact is the result or the output of a processing activity.	76%	12%	4%	8%
A processing applies property parameters of the performing component (i.e. cylinder surface).	62%	28%	0%	10%
The description of a function comprises input -> processing -> output as well as applied property parameters of the performing component.	62%	22%	4%	12%

Fig. 3. Comprehensibility of semantic context of function

All aspects were predominantly rated as well applicable. The participants had the opportunity to make announcements or proposals for improvement of the graphic contents. Conspicuous is the second aspect, which is well comprehensible on a very deep level of detail, where mostly a few or even one equation can express this effect. However, several participants remarked that this would cause high effort on the one hand and using equations will not be possible or suffice on low levels of detail on the other hand. Therefore, the level of accuracy should be limited to the necessary minimum for the applied context. Considering this condition, a kind of relation should always be possible to be specified between input and output. For instance, the highly complex system “combustion engine” could be sufficiently modeled by a characteristic map only using load and engine speed as input values and engine torque as output value, depending on the modeling purpose. Furthermore, the first statement (cf. Fig. 3) was criticized, because not every input triggers an activity. This statement has therefore been rephrased to “processing activities are triggered or excited by one or more certain inputs”. The schematic graphic only shows one input, which is transferred into one output. The meaning behind is not that stringent: an activity can have multiple inputs and multiple outputs. Moreover, the processing of the inputs may be conducted in a variety of ways, depending on the characteristics of the input values, which will become comprehensible through further decomposition of the activity. One participant stated that processing does not only apply parameters of the performing component, but also from adjacent systems.

Concluding, the term definitions as well as the remarks contributed to the advancements of the definition of a common language and associated semantic connections. The identified inaccuracies have meanwhile been revised in the specification. The common language forms the basis for the definition of a model implementation in SysML using extending profiles. Therefore, the application of SysML as a leading MBSE tool was assessed as well, combined with the identification of relevant modeling aspects for engineering tasks as well as improvement potential regarding SysML. The results are presented in the next chapter.

5 Application State of SysML and Current Advancement Issues

Firstly, the participants were asked to rate their own SysML experience. Only two out of 50 responses had no SysML experience and have therefore not answered the questions concerning SysML. 7 participants claim their selves as SysML experts, 5 as advanced modelers and 19 as modelers with basic experience. The remaining 16 participants have no modeling experience, but know SysML diagrams from literature.

Regarding application of the provided diagram types of SysML, the participants were asked to evaluate their particular benefit in representing the desired information of a modeled system. The results are illustrated in Fig. 4. The most frequently applied diagram type is the Internal Block Diagram for modeling internal system structures. Its benefit was rated as “crucial” by 40% of all participants. The benefit of Activity Diagrams was also rated as “crucial” by 40% at a little less application ratio. The most unknown diagram type is the Constraints Diagram, which is intended to represent constraints between model entities like parameters or requirements, merely 48% know this diagram type and only 4% rate its benefit as “crucial”.

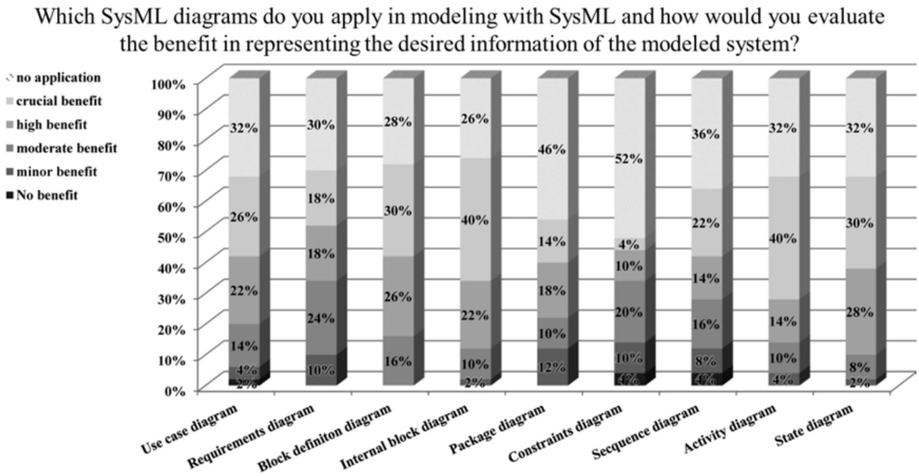

Fig. 4. Benefit of SysML diagrams for representation of particular modeling aspects

The next question demanded for the added value of SysML for modeling major tasks of discipline-crossing systems. In contrast to the provided diagram types, functional modeling has the most added value for users, which could indeed be conducted by IBD's, but would presume to apply according methods like the FAS-method (Functional Architectures for Systems, cf. [13], [14]), which applies this diagram type for enabling an additional view (containing functional blocks) in order to overcome the gap between solution-neutral modeling of activities and flows and the performing (physical) structure with interfaces in IBD's.

The following question asked for the importance of the previously mentioned aspects towards their general importance for the participant's work tasks. The comparison points out that the added value of SysML fits the user's need pretty well, but the results have shown that modeling of a system is much more demanded than data exchange between tools. Finally, the users had the opportunity to remark improvement potential regarding SysML in order to facilitate broader application of the modeling language in academic and industrial product engineering. 24 more or less detailed remarks were stated. Half of the participants remarked that not SysML itself should be improved, but rather the provided modeling tools, especially regarding usability (i.e. navigation through models, support of special views like matrices or special diagrams, handling etc.). Furthermore, six participants remarked missing modeling methods or guidelines and a high learning effort, five users missed particular aspects (i.e. decision tables, chances and risks). Insufficient Model2Model-transformation-support and variant modeling was also mentioned multiple times.

6 Conclusion and Outlook

The paper at hand has clarified that term understanding even among Systems Engineers in academia and industry is still very heterogeneous, but features tendencies towards corresponding aspects. This enables the opportunity to harmonize the understanding of basic terms like function and behavior in order to provide a basis to formalize those terms within modeling languages with according entities, attributes and relations. A graphical representation has been presented to the survey participants, which encountered predominantly positive responses. Hence, a formal specification of modeling elements can be derived incorporating minor advancements. Furthermore, the survey results have shown that SysML seems to be an adequate modeling language to cope with important modeling aspects supporting daily engineering work. Nevertheless, several advancements of SysML and in particular the modeling tools are still necessary in order to enable a wide application of Model-Based Systems Engineering in product development processes. Therefore, the IPEK conducts continuing high efforts in development of new, extending modeling aspects realizing the needs of product designers and managers (i.e. [11], [12]). Furthermore, a SysML extension for function-based modeling with derivation of dynamic structures through further implementation of the paradigms of the Contact & Channel – Approach (C&C^2-A) [10] is under development in order to obtain better acceptance among model users [15]. The long-term goal is to achieve more human-centered MBSE tools and methods.

References

1. Friedenthal, S., Moore, A., Steiner, R.: A practical Guide to SysML – The Systems Modeling Language, 2nd edn. Elsevier, Amsterdam (2011)
2. Tazir, N.: Viel diskutiert, selten wirklich angewendet - Systems Engineering. In: Produkt-Daten Journal (February 2011)
3. Bone, M., Cloutier, R.: MBSE Survey Results (Summary). Stevens Institute of Technology (2012)
4. Estefan, J.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. INCOSE MBSE Focus Group (2008), `http://omgsysml.org`
5. Kasser, J.: Seven Systems Engineering Myths and the corresponding Realities. In: Proceedings of the Systems Engineering Test and Evaluation Conference, Adelaide, Australia (2010)
6. Friedenthal, S.: SysML: Lessons from Early Applications and Future Directions. In: INCOSE INSIGHT, vol. 12(4). INCOSE (2009)
7. Karban, R., Hauber, R., Weilkiens, T.: MBSE in Telescope Modeling. In: INCOSE INSIGHT, vol. 12(4). INCOSE (2009)
8. Eckert, C., Alink, T., Ruckpaul, A., Albers, A.: Different notions of function: results from an experiment on the analysis of an existing product. Journal of Engineering Design (2011)
9. Vermaas, P.: Technical Functions: Towards Accepting Different Engineering. In: International Symposium Series on Tools and Methods of Competitive Engineering (TMCE), Ancona, Italy (2010)
10. Matthiesen, S.: Seven Years of Product Development in Industry – Experiences and Requirements for Supporting Engineering Design with ‘thinking tools’. In: Proceedings of the International Conference on Engineering Design (ICED), Copenhagen (2011)
11. Albers, A., Zingel, C.: Interdisciplinary Systems Modeling Using the Contact & Channel-model for SysML. In: Proceedings of the International Conference on Engineering Design (ICED 2011), Copenhagen, Denmark (2011)
12. Zingel, C., Albers, A., Matthiesen, S., Maletz, M.: Experiences and Advancements from One Year of Explorative Application of an Integrated Model-Based Development Technique Using C&C^2-A in SysML. IAENG International Journal of Computer Science 39(2) (2012)
13. Lamm, J.G., Weilkiens, T.: Functional Architectures in SysML. In: Proceedings of the Tag des Systems Engineering (TdSE), Munich, Germany (2010)
14. Korff, A., Lamm, J.G., Weilkiens, T.: Werkzeuge für den Schmied funktionaler Architekturen. In: Proceedings of the Tag des Systems Engineering (TdSE 2011), Hamburg, Germany (2011)
15. Albers, A., Zingel, C.: Extending SysML for Engineering Designers by Integration of the Contact & Channel – Approach (C&C^2-A) for Function-Based Modeling of Technical Systems. In: Proceedings of the Conference on Systems Engineering Research (CSER 2013), Atlanta, Georgia, USA (accepted 2013)