NordDesign 2016
August 10 – 12, 2016
Trondheim, Norway

Supporting SysML model generation in early phases of the development process

Johannes Kößler, Kristin Paetzold

Institute of Technical Product Development, Universität der Bundeswehr München
Johannes.koessler@unibw.de, kristin.paetzold@unibw.de

Abstract

Within the development of technical systems more and more domains and needs from different stakeholders have to be integrated. Based on that, the Model-based Systems Engineering is becoming more interesting for companies. But the integration of this approach within the existing development process is a difficult task. SysML as a result of the approach of MBSE is a modeling language. It enables the developer to model and visualize a technical system in respect to different perceptions. Currently SysML is mainly being used for the requirements management and their organisation. This paper shows an initial approach, how the integration of more SysML models into the early phases of the development process can be supported. The main goal is to show how existing methods and their results can be used to support the generation of SysML models. Thereby it focuses on the generation of block definition diagrams. These diagrams can serve as a starting point of a more general integration of SysML. Concentrating on a single type of diagrams the integration is accelerated and simplified. In addition not every possible piece of information, which could be stored within block definition diagrams is used. Mainly Meta information based on previous development tasks and results of existing methods is stored in the block definition diagrams. These restrictions limit the amount of data which then can be mapped to block definition diagrams. Based on the mapped data the paper shows possible integrations of the SysML diagrams into the development process. These diagrams can serve as a visualisation of a preliminary structural definition of a system. In addition they can be used for controlling subsequent tasks and structuring the data and information flow.

Keywords: SysML, Development process, MBSE, Early phases, System integration

1 Introduction

The product development is faced with an increasing complexity. This is a result of several reasons, for example the increasing functionalities a technical product must meet and the increasing number of domains working together to develop a product.

1.1 Problem description

To support the development of technical products it is not absolutely necessary to develop new methods, but to introduce already existing ones. During the last decades the product validation with simulations has become more and more important. Especially in the early phases of the development process the use of simulations is important to ensure the fulfilment of the requirements. Therefore new approaches have been developed to support these challenges of the development processes. One of these approaches is Model-based Systems Engineering, which tries to support the development by focussing on product models starting in the very beginning of the development process. Therefore, the modeling language SysML has been created. The developer is allowed to model a technical system with different perceptions. But integrating this new approach and the modeling language SysML into existing development processes is still an open task. It needs additional tools for the modeling of the system and also new methods which have been introduced by Model-based Systems Engineering. Until now this integration has only partially happened and is currently a topic many researchers are working on. In particular the integration of SysML-models is not yet completed neither into the processes nor into the data and information flows (e.g. PDM-Systems).

1.2 Objective

The main objective of this paper is the support of the generation of SysML models during the early phases of the development process. SysML is a powerful modeling language introduced by the approach of Model-based Systems Engineering. It offers many possibilities to support the development process. But it is a very difficult task to use in existing development processes. Therefore this paper describes a possibility of how the SysML modeling can be supported by already existing methods and tools used during the development process. The support focuses on a development task at the very early phases of a development process. In this phases the complexity of the technical system is lower compared to the later phases. In addition the development task is a development of an alternative version of a single component or a smaller assembly. Based on this restriction existing knowledge can support the development and the number of domains included in the development is also lower than the development of a bigger technical system. It is not trying to integrate SysML-models into bigger development tasks like the development of a complete car nor to integrate SysML models into a complete development process.
The main research questions answered in this paper are:

- How can the generation of SysML-models be supported by methods used in early phases of the development process?
- How can these generated models be integrated into the development process?

Based on the main research question additional questions are:

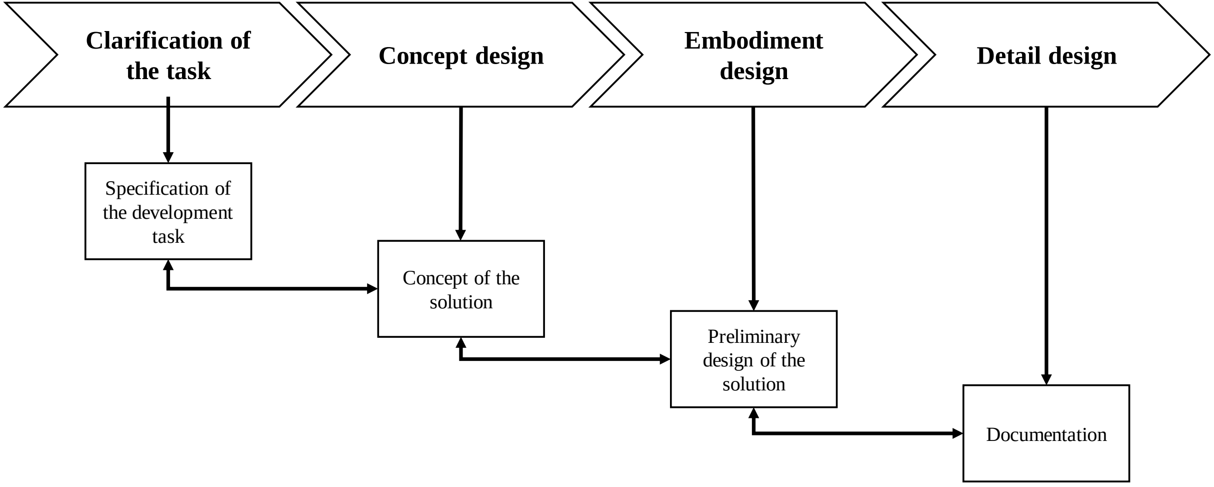
- Which data can be used to support the model generation?
- Which type of SysML diagram can be used in the early phases of the development process to serve as an initial use of SysML?

2 State of the art and related work

This section describes the topics which serve as a base for this paper. First of all two exemplary approaches of the development process are described. They show how the product development could look like and explains some challenges the product development is faced with. The next sections briefly describe the approach of model-based systems engineering and its advantages including the modeling language SysML.

2.1 Development processes

Many different approaches have been developed to describe and support the development processes. Generally all of them start with the requirements a technical system should fulfil. Also all of them try to structure the complete process; they have different ways of how to define the phases of the development process.
A very common approach is the development process by Pahl/Beitz (Pahl et al., 2007). This approach basically defines the development process with four different phases.



Clarification of the task
Concept design
Embodiment design
Detail design
Specification of the development task
Concept of the solution
Preliminary design of the solution
Documentation

Figure 1. Simplified approach of Pahl/Beitz (Pahl et al., 2007).

As shown in figure 1 this approach starts with the process step of clarification of the task in which the development task is specified. During each step more information is generated and the description of the product is getting more detailed. This approach also describes methods that are supporting each process step as well as the results of each process step. After achieving the described results of each process step the next step can be started. In addition this process allows to step backward if it is necessary. But this process description mainly focuses a single domain development task (e.g. a mechanical system) and does not support development processes which include many different domains. In particular the task of system integration of the results of different domains is not a part of this approach.
As this paper is focussing on the very early phases of the development process this phases are mainly during the process step clarification of the task. The beginning of the process step concept design is also included but not the complete process step.

Another approach is the CPM/PDD approach developed by Weber (Weber, 2005). This approach only uses two development steps. Compared to the approach of Pahl/Beitz, which focuses on the process phases, the approach of Weber concentrates on the data generated during the development process. Thereby Weber supports the classification of the Data. The following figure shows the approach of Weber with a simplified visualisation.

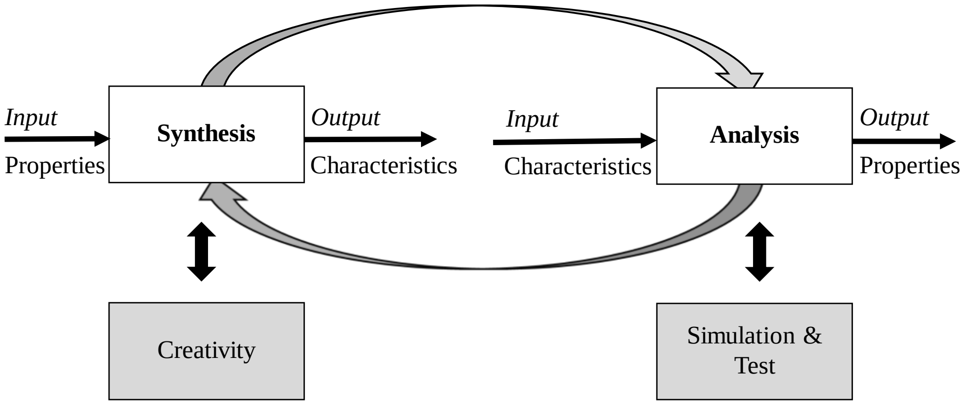


Figure 2. Simplified approach of Weber (Weber, 2005).

Weber uses just two process steps which are synthesis and analysis. The process step of synthesis is a step of creativity, in which the developer is trying to find characteristics. The characteristics should result in a solution of a system, which fulfils the requirements. The other step is the process step of analysis. During this step the developed characteristics are used to test or simulate the system and evaluate, whether the properties of the system are fulfilling the requirements or not. The two steps are being repeated, until the requirements are completely fulfilled or the deviation is within defined ranges. Weber additionally considers external factors, which are not included in figure 2. Within the two process steps Weber uses two different types of data. These are the characteristics and the properties. The characteristics describe the structure, shape or material of a product and can be influenced by the developer. The properties define the behaviour of a product and cannot directly be influenced by the developer, but they are a result of the characteristics. By comparing the requirements with the achieved properties the target achievement can be measured.

During the very early phases neither real tests nor detailed simulations are possible. But parameter based simulations for a property validation are possible. For those kinds of tests models are needed which support the generation of simulation models.

Beside these two approaches many others are existing, for example the concept of Hubka and Eder (Hubka et al., 1992) or the Axiomatic Design Theory by Suh (Suh, 1990, 2001). Naming all of them would go beyond this paper.

These general approaches serve as a basis for companies in the industry. But they often use only parts of those general approaches and adapt them to their needs. This results in an uncountable number of different development processes. Each of them is supported with different tools and methods for supporting single process steps and tasks during the development. In particular the software tools for the development (e.g. CAD), the evaluation (e.g. simulation tools) and the management of the data (e.g. PDM-System) lead to a very heterogeneous development environment. This results in the need of supporting the integration of general approaches into the development processes.

2.2 MBSE

As already mentioned the product development is faced with a very heterogeneous environment in terms of the used tools and methods. The approach of Model-based Systems Engineering (MBSE) tries to partly solve this situation. The approach of MBSE is supporting the entire life cycle starting from the requirements until the end of the development process (INCOSE, 2007). This should be achieved by representing each technical system with system models. These models support the entire development process (Weilkins, 2008) and also consider the needs of different stakeholders (Anderl et al., 2012).

By formalizing the use of system models MBSE is transferring the storage of data from documents into system models. It also tries to integrate different domains into the models to make them able to better work together. An example of the different models used in MBSE is given in the following figure.

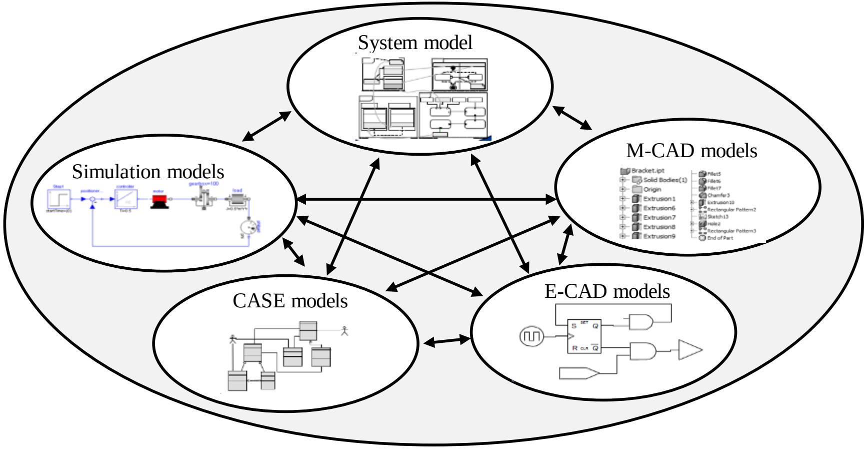


Figure 3. Model-based product development (Eigner et al., 2014)(Friedenthal et al., 2008).

MBSE considers two different perceptions of a system description. These are the structural-based and the functional-based views. The structural-based view divides the complete system into subsystems, assemblies and components. The system is then built up by hierarchically combining these components. The functional-view describes the complete system by combining its sub functions to represent the functionality of the complete system.
Currently many approaches are existing, which are implementing MBSE, e.g. MBSE approaches of NASA (Nasa, 2007), OOSE (OOSEM, 2006) and Vitech (Estefan, 2007).

2.3 Systems Modeling Language (SysML)

As a result of the development of the approach OOSE the modeling language SysML as an advancement of UML has been developed. This language allows the modeling of a system in consideration of the functional and the structural view. First approaches to integrate SysML into the development processes are currently being developed (Augenbaugh et al., 2004)(Eigner, 2012). But the integration of this language into the development processes is not yet completed.
SysML offers different types of diagrams to model a system, the taxonomy is shown in the following figure.

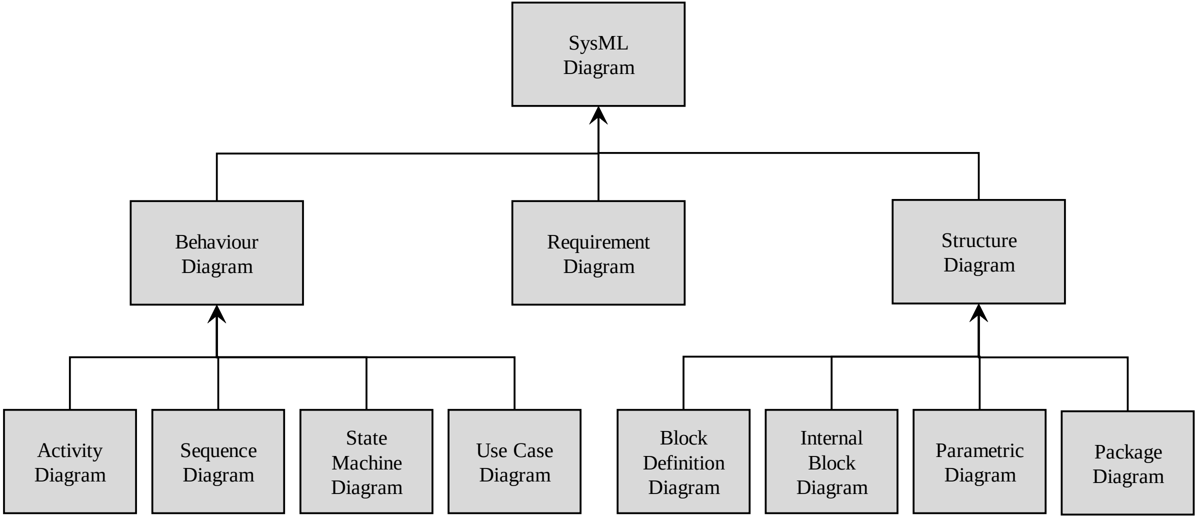


Figure 4. SysML diagram taxonomy (Friedenthal et al., 2009).

SysML mainly offers three different types of diagrams. These are Behaviour diagrams, Requirement diagrams and Structure diagrams. This paper is using the diagram type Block Definition Diagram. As we are focusing on the early phases of the development process the level of detail is relatively low and the use of additional diagrams is not necessary.

3 Generation of SysML models in early phases

The generation of SysML models requires a lot of time and knowledge about the system that has to be modeled. Especially the amount of time needed for the generation is a big challenge. First studies based on interviews of developers and system engineers have shown, that this is one of the critical challenges while introducing SysML into the development process and could also potentially avoid the use of SysML. Thereby the support of the actual model generation is a possibility to support the general use of SysML.

The most important task during the integration of SysML into the development process is the connection of the SysML models and tools with the existing environment of the development process. This connection is necessary to support an efficient integration and use of SysML models. Some tools which are needed for this integrations are:

- PDM Systems
- PLM Systems
- ERP Systems
- Simulation tools like (Matlab/Simulink or Modellica)

There is a lot of research currently being done, which is addressing this integration (Augenbaugh et al., 2004)(Eigner, 2012). Depending on the development task and the current phase of the development process the available information, which can be used for the model generation, varies. During the early phases of the development task the available information is relatively low. But there is a big variety of methods that supports this phases.

Some of these methods are (Lindemann, 2009):

- Functional modeling
- Morphological boxes
- ABC analysis
- Similarity analysis
- Brainstorming

All of these mehodes are established and currently supporting the development process. They are used in different situations and with different purposes. But their results can often and directly be used to generate SysML models. Or at least offer additional information which can be integrated into existing SysML models. Thereby they can support the generation of SysML models and supporting the development process by allowing an evaluation of the results of the methods. Two examples are given in the following.

Use of functional modeling:

As a result of the functional modeling information about the functions of the system are available. Based on this data a SysML model can be generated which includes the different functions and their connections. In this situation SysML can be used to visualise the different connections and to give a hint which connection could be crititcal (e.g. are different domains working on one connection?).

Use of morphological boxes:

The result of a morphological box is a selection of different solutions for each function listed in the morphological box. By selecting one solution for each function different solutions for the system can be generated. These different selections of a systems can also be used for generating a SysML model. The SysML model can be evaluated and the results can help to compare the different solutions.

An example of a SysML model generated based on a functional modeling is shown in the following figure.

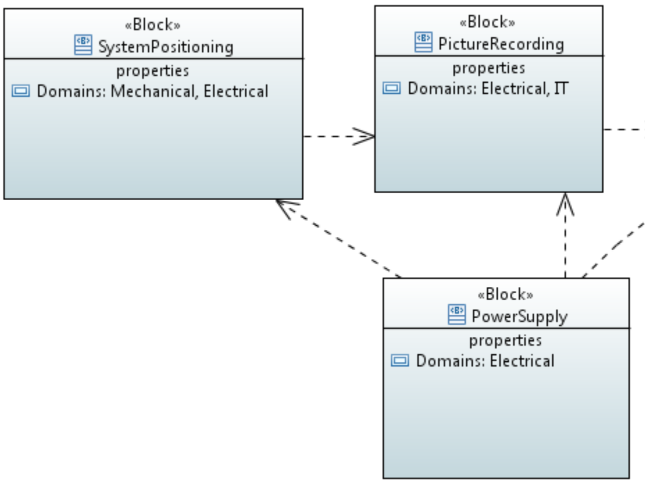


Figure 5. Example of a functional structure for cargo securing on a road vehicle.

This example shows four functions which are needed for controlling the cargo securing on a road vehicle. It only contains the information about the functions, the domains involved and the connections between the functions.

These models can be used to evaluate the interfaces between the functions of the system. By automatically analyzing the different solutions the SysML model allows to compare the different solutions and supports their evaluation. Some possible evaluations are:

- Quantity of interfaces between different domains
- Quantity of different domains

Depending on the additional information a lot more evaluations are possible, for example:

- Quantity of purchase parts
- Quantity of different solutions of each function

As shown typical methods of the early phases of the development process can support the generation of SysML models. Based on the low level of detail the method results can be integrated into these models and support the process with basic evaluations. During the following phases of the development process these models can serve as a base for more detailed models.

4 Integration into the development process

Beside the generation of SysML models the integration and use of them during the development process is another important task. Based on the generation of models as described in chapter 3 their use within the development process is described in the following.

The main goal of the process integration is the support of a generation of a system model and the evaluation of different possible solutions. The focus thereby is on a functional description of the system.

The general integration of the system model generation into the early phases of the development process is shown in the following figure.

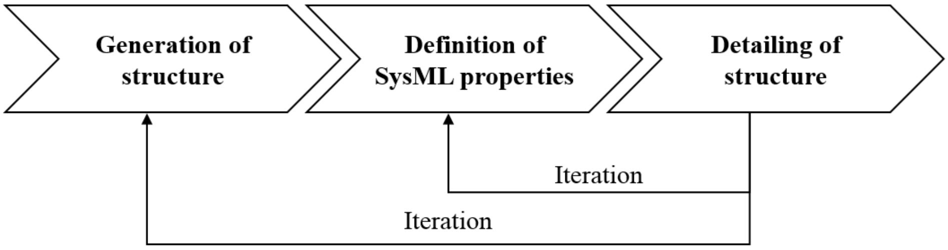


Figure 6. Procedure for integrating and detailing SysML models during the early phases.

The integration mainly consists of three different steps which are:

1. Generation of the structure
2. Collection of necessary information
3. Detailing and evaluation of a solution

First of all the functions of the system have to be identified and structured. As a result of this step a first system structure can be modeled. Beside a general description of each function the connection between these functions are necessary for the SysML model. During the early phases this information can be generated by methods like the functional modeling.

The next step is a task of data collecting. During this step the data has to be collected which is needed for the organization of the development process. The goal of this step is to generate a template for definition of each function within the SysML model. This depends on the general organization of the development process of each company, on the development task and the tasks that have to be done during the early phases. Some examples are:

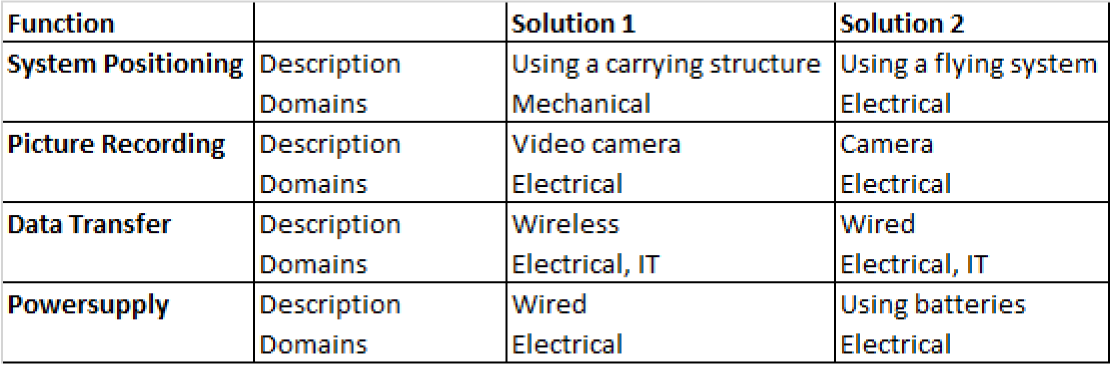
- If there is more than one domain involved each function will need information about the domain.
- If a function is critical it will be necessary to evaluate different solutions.
- Is the task a development task of a new product?

Currently we are working on a questionnaire which will support the identification of information that has to be stored in the SysML models.

At the end of the second step the SysML model with the functional structure of the system has been extended with additional information about each function. This function has to be integrated into the SysML model.

During the third step the extended SysML model can be used to generate templates for additional development tasks. The results of these tasks serve as additional data which can be stored within the SysML model. In addition it can be used for the evaluation of different solutions as explained in chapter 3.

An example of the support of a development task is shown in the following picture.



Function		Solution 1	Solution 2
System Positioning	Description	Using a carrying structure	Using a flying system
	Domains	Mechanical	Electrical
Picture Recording	Description	Video camera	Camera
	Domains	Electrical	Electrical
Data Transfer	Description	Wireless	Wired
	Domains	Electrical, IT	Electrical, IT
Powersupply	Description	Wired	Using batteries
	Domains	Electrical	Electrical

Figure 7. Example of a generated morphological box.

Based on the system structure a template of a morphological box has been generated. It has to be filled and in addition can be used for an evaluation of the different solutions. Depending on the evaluations which have to be done the layout and the data stored within the template can change. The given example just uses information about the the involved domain, a possible supplier, the know-how about the solution and a description.

5 Discussion

This paper describes a possibility of a generation of SysML models during the very early phases of the development process. In addition it describes how these models can be integrated into the development process.

The main idea is not to use existing systems like PDM, PLM or ERP systems for the generation. Instead of the integration of these tools it describes the use of certain methods and their results for the generation. This is briefly shown with the use of a functional modeling and morphological boxes. The integration of these models into the development process is also shown with the use of these methods. The actual support of the model generation as well as the integration of them into the development process is not achieved by an integration of tools. But it is achieved by the use of SysML templates and templates for the different methods used during the early phases.

Currently this initial approach is focussing only on a few methods used in the development process. The next steps are the generation of a questionnaire which will support the identification of information that can be used within the SysML models and the integration of more typical methods used during the early phases. The approach also has to be expanded to use these models during the entire development process and not only during the very early phases.

6 Conclusion and Outlook

The main topic of this paper is the question how the generation of SysML models and their integration into the development process can be achieved. Thereby it is focussing on the very early phases of the development process. During these phases it is considering the generation of possible solutions and their evaluation. With these limitations it is recommending the use of existing methods to generate the necessary data for the generation of SysML models. In addtition it shows how these models can be integrated into the development process also focussing on the early phases of the development process.

The next tasks will be the extension of the described initial approach of using and integrating SysML models. Currently we are working on a system model which will support not only the early phases but also the later phases of the development process. Additionally we are integrating more methods to support the SysML model generation.

Citations and References

Anderl, R., Eigner, M., Sendler, U., Stark, R. (2012). Smart Engineering – Interdisziplinäre Produktentstehung – acatech DISKUSSION.

Aughenbaugh, J.M., Paredis, C.J.J. (2004). The Role and Limitations of Modeling and Simulation in Systems Design. *ASME International Engineering Congress, Anaheim, USA.*

Eigner, M., Gilz, T., Zafirov, R. (2012). Neue Methoden, Prozesse und IT Lösungen für die virtuelle disziplinübergreifende Produktentwicklung. *Berns, K.; Schindler, Christian;*

Dreßler, K; Jörg, B.; Kalmar, R.; Zolynski, G. (Hrsg.): "Proceedings of the 2nd Commercial Vehicle Technology Symposium 2012".

Eigner, M., Roubanov, D., Zafirov, R. (2014). Modellbasierte Virtuelle Produktentwicklung. *Springer Verlag, Berlin Heidelberg.*

Estefan, J. A. (2007). Survey of Model-Based Systems Engineering (MBSE) Methodologies. *Technical report, California Institute of Technology, May 25 2007.*

Friedenthal, S., Greigo, R., Sampson, M. (2008). INCOSE MBSE Roadmap. *In „INCOSE Model Based Systems Engineering (MBSE) Workshop Outbrief", S. 6. Paper presented at INCOSE InternationalWorkshop 2008, Albuquerque, NM, 26. Jan 2008.*

Friedenthal, S., Moore, A., & Steiner, R.. (2009). A Practical Guide to SysML - The Systems Modeling Language. *Morgan Kaufmann OMG Press, Burlington, MA.*

Hubka, V., Eder, W.E. (1992). Einführung in die Konstruktionswissenschaft. *Springer, Berlin-Heidelberg, 1992.*

INCOSE. Systems Engineering Vision 2020. *Technical Operations International Council on Systems Engineering (INCOSE).*

Lindemann, U., (2009). Methodische Entwicklung technischer Produkte – Methoden flexible und situationsgerecht anwenden. *Third edition, Springer Verlag, Berlin Heidelberg.*

NASA/SP-2007-6105 (2007). NASA Systems engineering Handbook.

OOSEM (2006). Object-oriented Systems Engineering Method. *OOSEM Descriptive Qutline for INCOSE SE Handbook Version e, Annoted update.*

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H. (2007). Engineering Design: a systematic approach. *Third edition, Springer-Verlag, London, UK.*

Suh, N.P. (1990). The Principles of Design. *Oxford University Press.*

Suh, N.P. (2001). Axiomatic Design. *Oxford University Press.*

Weber, C. (2005). CPM/PDD - An extended theoretical approach to modeling products and product development processes. *Proceedings of the 2nd German-Israeli Symposium on advances in methods and systems for development of product and processes. TU Berlin / Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK), 07.-08. Juli 2005, p.159-179, Stuttgart, Fraunhofer-IRB-Verlag.*

Weilkins, T. (2008). Systems Engineering mit SysML/UML - Modellierung, Analyse, Design. *dpunkt Verlag, 2nd. Edition.*