

Marvin M. Schmidt, Rainer Stark (2020): Model-Based Systems Engineering (MBSE) as computer-supported approach for cooperative systems development. In: Proceedings of the 18th European Conference on Computer-Supported Cooperative Work: The International Venue on Practice-centred Computing on the Design of Cooperation Technologies - Exploratory Papers, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.18420/ecscw2020_ep04

Model-Based Systems Engineering (MBSE) as computer-supported approach for cooperative systems development

Marvin M. Schmidt¹, Rainer Stark^{2,1}

¹Fraunhofer IPK, Germany, ²TU Berlin, Germany

Marvin.Michael.Schmidt@ipk.fraunhofer.de

Abstract. With rising globalization and a trend towards Cyberphysical systems (CPSs) as well as smart products the demand for cross-company and interdisciplinary collaboration increases. To handle the complexity of these systems and products Model-Based Systems Engineering (MBSE), as an enhanced form of Systems Engineering (SE), has emerged in engineering and is adopted by many companies. While this approach tries to cope with the current complexity trends, it does address the collaborative aspect of product creation only in a small scope. This paper shall address the combination of MBSE and collaboration in engineering to form a computer-supported approach for collaborative systems development.

Copyright 2020 held by Authors, DOI:10.18420/ecscw2020_ep04

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, contact the Authors.

Intro

Products have become more complex in recent years (Lindow et al., 2011). For coping with the rising complexity, companies establish joint ventures as well as other forms of cooperation and use development approaches for product creation that are more sophisticated, covering modern processes, methods and tools.

As most of the modern products can be seen as 'systems', which in the sense of engineering are defined in (ISO 15288, 2015, p. 9) as a 'combination of interacting elements organized to achieve one or more stated purposes', Model-Based Systems Engineering (MBSE) is one of the approaches that is gaining more attention in recent years. Even though, Systems Engineering (SE) as foundation of MBSE is meant to be ' [...] a transdisciplinary and integrative approach [...]' (INCOSE, 2019), it's current state is not yet as transdisciplinary and integrative as it should be. Huldt and Stenius (2018) for example pointed out that the system engineering tools used for SE tasks do not connect and integrate different domains and thus keep the benefits of MBSE inside the SE domain.

Goal of this paper is taking a closer look at MBSE as a computer-supported approach for collaborative systems development. It shall give a brief overview of MBSE as well as cooperation and collaboration in general and then focus on some current challenges as well as possible solutions for these challenges. These possible solutions shall give some research topics for further investigation.

State of the Art

Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE) has yet no internationally converged definition (Huldt and Stenius, 2018). The most widely used definition is from the International Council of Systems Engineering (INCOSE), is one of the largest organizations focusing on systems engineering and has defined MBSE as:

'[...] the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.' (INCOSE Technical Operations, 2007, p. 15)

This definition covers multiple aspects. The mentioned activities and the consideration of all life cycle phases refers to the SE-part of the term MBSE, which focuses on enabling the realization of a system from stakeholder needs to the final system or solution. While the activities described in the mentioned definition have been done document-based in the conventional SE, the 'Model-Based' aspect of the MBSE definition means to use models as artifacts for these activities. While models have been used in document-based SE for specific

aspects and analysis, MBSE aims to create a holistic model of the overall system that integrates the various models within itself and the various modeling activities, mentioned in the MBSE definition above, into SE (Friedenthal et al., 2012, p. 45). This model-based approach, which is used in some specific domains like software engineering (e.g. Unified Modeling Language (UML) models) and mechanical engineering (e.g. Computer-aided Design (CAD) models) for quite some time, is meant to replace the document-based approach (INCOSE Technical Operations, 2007, p. 15), as it has multiple benefits (INCOSE, 2015, p. 189):

- multiple perspectives on the system model and thus an easier analysis
- improved product quality due to consistency, correctness and completeness evaluation possibilities
- knowledge reuse by using standardized capturing of information
- possible reduction of cycle times due to standardization and reuse
- easier maintenance and synchronization of information compared to document-based approaches.

The holistic model of the overall system is called system model. It '[...] consists of model elements that represent requirements, design, test cases, design rationale, and their interrelationships' (Friedenthal et al., 2012, p. 17). There are different approaches to connect the elements of the system model. Stark and Schulze (2010) pointed out three forms of linking system model elements, presented in figure 1:

1. inter-linkage between system elements
2. inter-linkage through a meta-model
3. inter-linkage of partial system results

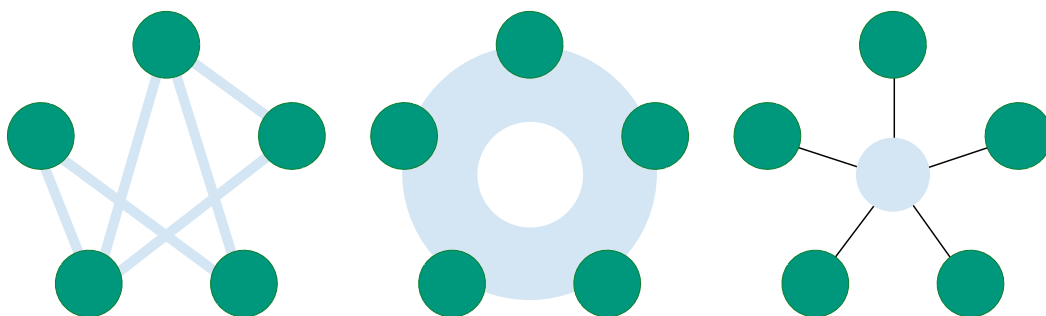


Figure 1. System modeling approaches presented by Stark and Schulze (2010) left to right: inter-linkage between system elements, inter-linkage through a meta-model, inter-linkage of partial system results.

An example for the inter-linkage between system elements is a direct link and exchange between a requirement, a test case for the verification of that requirement and a component inside the product structure that needs to fulfill that requirement.

To allow this link a standardized interface is required. In that way, this approach allows a tool-independent linking and usage.

The inter-linkage through a meta-model is the typical approach of modeling tools like PTC Windchill Modeler, Eclipse Papyrus, IBM Rational Rhapsody or Cappella. They use modeling languages like Systems Modeling Language (SysML) to model various elements and store them in a model repository (Haskins, 2011). These tools create and manage a meta-model in their backbone, which stores the interconnections of these modeled system elements.

The third form presented by Stark and Schulze (2010) focuses on the inter-linkage of results. In that way, the stakeholders can model and simulate all partial system elements independently and the other system elements solely interact with their results. Therefore, key parameters have to be linked dynamically and the results of, e.g. the domain-specific simulations, can be exchanged, without influencing the internal modeling of other system elements and without unwanted exposure of internal knowledge. An example is the Smart Hybrid Prototyping (SHP) approach of Auricht et al. (2012), where they have linked results of the behavior model to the output of a haptic feedback device and to a Digital Mock-Up (DMU), to allow an interaction with a physical prototype.

The mentioned inter-linkage between the model elements can be associated with the term traceability, which has been adapted from the software industry. Traceability is defined in ISO 29110 (2016), which is currently in publication, as description of an '[...] association among two or more logical entities, such as requirements, system elements, verifications [...], or tasks [...]' (ISO 29110, 2016, p. 10). It has been topic in multiple research work, as for example Königs et al. (2012) and Sünnetcioglu et al. (2017). They all deliver feasible approaches to enhance traceability but focus only meta-models and tool integrations for traceability handling. In contrast, Eckl et al. (2015) uses SysML as modeling language for a meta-model but does rely on Open Services for Lifecycle Collaboration (OSLC) as well to allow an inter-linkage of model elements with domain specific models like CAD models through a standard interface. We value the tool-independent approach as more promising, as there already exist approaches for semantic web in other domains, as for example Kim and Chung (2005), Jacobs et al. (2014) as well as Wang et al. (2018). For further research, this tool-independent approach shall be favored.

As MBSE has been defined as '[...] formalized application of modeling [...]' (INCOSE Technical Operations, 2007, p.15) in the definition above it is neither a process nor a methodology in itself. To give the users of MBSE some kind of guidance for the application of MBSE multiple methodologies have been developed. Methodology means a combination of methods, processes and tools used for that objective (Estefan, 2008). Estefan (2008) list IBM Telelogic Harmony-SE, INCOSE Object-Oriented Systems Engineering Method (OOSEM), IBM Rational Unified Process for Systems Engineering (RUP SE) for Model-Driven Systems Development (MDS), Vitech MBSE Methodology, JPL State Analysis (SA) and Object-Process Methodology (OPM) as examples of

current methodologies. Roques (2016) additionally presented the Architecture Analysis and Design Integrated Approach (ARCADIA) method with its Domain Specific Modeling Language (DSML). Until today none of these methodologies has emerged as standard in MBSE.

As we want to avoid tool-dependency, a rough overview on the basic activities in MBSE shall be given using the V-model of Fraunhofer IPK and TU Berlin as a possible process approach. It is based on the V-model as used in the VDI 2206 (2004) methodology for mechatronic system development and has been further developed by Beier et al. (2014) on the left side V-model and Buchholz et al. (2018) by forms of prototypes on its right side. Figure 2 depicts the adapted V-model presented by Buchholz et al. (2018).

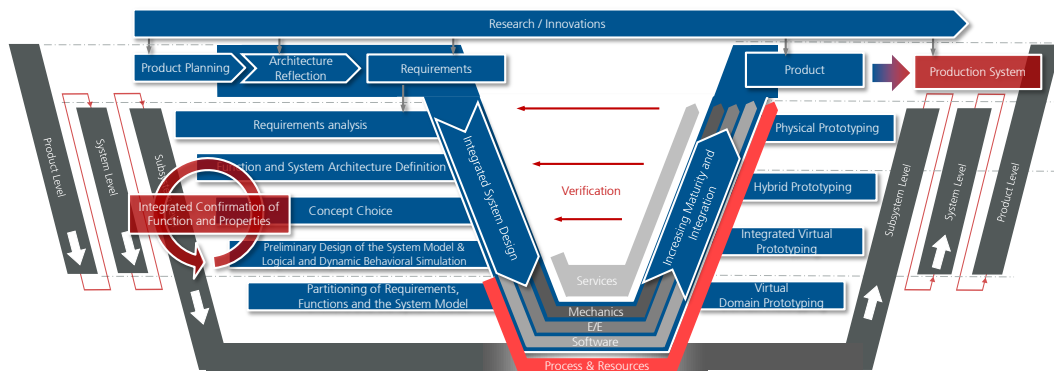


Figure 2. MBSE-process model presented in Buchholz et al. (2018).

The MBSE-process model, presented in figure 2, considers the complete development procedure from product planning as first step to a product with respective production system as output. According to Beier et al. (2014) and Buchholz et al. (2018), the process comprises the following activities:

1. **Product definition:** (*Product level only*) This point includes the first 3 aspects from 'Product Planning' through 'Architecture Reflection' to the 'Requirements'. While most development processes, as for example VDI 2206 (2004), start with the requirements as input, this approach considers them as output of previous tasks.
2. **Requirement Analysis:** (*all levels*) The product requirements defined in the previous phase have to be analyzed and checked for formal correctness and quality. In MBSE this is done with an authoring tool that allows to store the requirements in a formal requirements model and possibly automate this analysis.
3. **Function and System Architecture Definition:** (*all levels*) With the correct requirements, the functional architecture and the system architecture are defined. These are structured combinations of either functions or systems in a decomposed form based on the requirements. An example could be 'store

energy' as a function and 'photovoltaic power station' as a system for an requirement 'the system shall store 20,000kWh of energy generated by an photovoltaic power station'. On lower system levels of the development procedure, the system architecture can include components as well.

4. **Concept Choice:** (*all levels*) In this phase, the system architecture and the functional architecture have to be mapped to each other. As the 'photovoltaic power station' of the previous system example cannot be mapped to the function 'store energy', alternative and additional systems and solutions have to be considered, e.g. 'battery system' and 'hydrogen storage'. These have to be integrated into the system structure, checked for incompatibilities and mapped to the respective functions.
5. **Preliminary Design of the System Model & Logical and Dynamic Behavioral Simulation:** (*all levels*) In this phase, the system model is designed by connecting the functions, components and a related behavior with each other, and then simulated to test the logical and dynamic product, system and subsystem behavior. The key of this step lies on checking if some aspects are missing or have been modeled badly. Additionally, it can be used to examine first rough parameters for the chosen concepts. When the simulations have been executed successfully and delivered the expected results, the procedure is started over from step 2 for the system and for the subsystem levels.
6. **Partitioning of Requirements, Functions and the System Model:** (*Subsystem levels*) When all phases have been gone through for the product, system and subsystem level, a partitioning of requirements, functions and the system model in general is done on the subsystem level as lowest level in this process. Common domains for the partitioning are Mechanics, Electronics and Electrics (E/E) and Software for mechatronic systems, expanded by services as well as process and resources in the shown process, to make this process applicable to the previously presented systems as well. In the domain-specific design, each domain is working on its own solution within given boundary conditions and with the respective interfaces defined by the system model.
7. **Prototyping:** (*various levels*) In the integration phase on the right side of the V-model, various prototypes are created. Each domain designs virtual prototypes as last step of the domain specific design phase. These prototypes are continuously integrated until the interaction can be tested on a physical prototype. In this way, the product can be tested virtually considering all development aspects.

This process is iterative, which means the activities influence each other and are refined in the development process. The users has to chose the form of modeling these activities. It can be used full tool suites like IBM Rational or PTC Windchill

with tools integrated tools e.g. for requirements management (2) and modeling of use cases and system architectures (3) or independent tools for the activities, e.g. Eclipse Papyrus for the architecture definition (3). Same goes for possible modeling languages like SysML, MECHATRONICS UML (Schäfer and Wehrheim, 2010), that tried to already address the mechatronic system design or the language of Systems Modeling & Management Tool (SysMT), in which Königs (2012) tried to address some weaknesses of SysML. This is the foundation for an integrated MBSE approach. Independent of the used tools and modeling languages a overall system model has to be created in one the forms presented by Stark and Schulze (2010). We want to avoid the limitation to a specific modeling language or tool and therefore consider these references as valuable input but do not use one of them exclusively.

A rather new approach of handling systems engineering is the Advanced Systems Engineering (ASE). In addition to the previous aspects of SE and MBSE, ASE '[...] gives human considerations a central place in the integrated design and development of systems throughout their whole life cycle' (Albers and Lohmeyer, 2012, p. 413). This human-centered development is an important topic in today's research and should be considered in further research on this topic.

Collaboration and cooperation in engineering

Cooperation and collaboration are often mixed in terminology, while actually addressing a different depth of working together. Cooperation is '[...] the practice of people (or greater entities) working in common with shared resources and methods, instead of working alone or competitively' (Lu et al., 2007, p. 22). Collaboration '[...] aims at achieving a common goal and collective results that individuals would be incapable of accomplishing alone. In other words, collaboration requires a team of individuals to work on tasks that not only have shared resources (as in coordination) and shared outcomes (as in cooperation), but, most importantly, a shared common goal' (Lu et al., 2007, p. 22). As MBSE is focusing on the realization of one or more successful systems, there is a common goal that cannot be accomplished by an individual and thus highlights the required consideration of the collaboration aspect of MBSE.

Even though MBSE is a transdisciplinary approach for handling the realization of complex systems and INCOSE (2015) states the integration of all disciplines as one mayor point of SE, MBSE does not handle collaboration by definition. INCOSE Technical Operations (2007) even pointed out that

'Collaboration support capabilities for pervasive high performance, geographically distributed teams, would complement the integrated process support environment above by providing more empowerment for individual systems engineers and better collaboration support for performing team activities.' (INCOSE Technical Operations, 2007, p. 29)

New approaches like ASE try to address this topic by focusing more on the human (Albers and Lohmeyer, 2012). The computer-supported cooperative work (CSCW) community addressed the problems of cooperative work for a long time. Wang and Tang (2005) focused on collaborative work with computer-aided technologies (CAx) software, mainly for product design. Jia and Zhang (2007) developed a Web Service to integrate control model, mechanical model and hydraulic servo model. More recently, Cyberphysical systems (CPSs) as defined by Tomiyama et al. (2019) have been topic of CSCW research, e.g. in De Carvalho et al. (2018) and Hoffmann et al. (2019). They investigated CPSs as a mean for knowledge and expertise sharing in industry. Wouters et al. (2017a) evaluated their observations concerning a collaborative systems engineering scenario and pointed out five challenges:

1. A shared vocabulary is required. This challenge is based on the observation that the stakeholders have to share their data, which implies an agreed vocabulary.
2. Orchestration has to be provided to manage sequential tasks and inter-dependencies.
3. The Exposition of data has to be controlled meaning that not everything has to be exposed and most stakeholders want to keep their knowledge due to industrial know-how protection.
4. Consistent data is required and constrained by domain rules, which means that the data is heterogeneous due to the various domains.
5. The collaboration is typically part of a network of collaborations and has to be handled as such.

In Wouters et al. (2017b) they addressed the challenge of a shared vocabulary by semantic projection of one stakeholder's data to another stakeholder and thus allowing a common understanding, while each stakeholder may use its own vocabulary. The other challenges, that have been been investigated in their prior work, have not been addressed.

Lu et al. (2007) investigated similar problems in their work and comprised them to three main challenges: shared definition of the problem (common understanding), requirement for a same set of information and the understanding of rules for decision making. They have developed a framework for collaborative engineering based on investigations made in industry. Lu et al. (2007) define collaborative engineering as 'a new sociotechnical engineering discipline, which facilitates the communal establishment of technical agreements among a team of interdisciplinary stakeholders, who work jointly toward a common goal with limited resources or conflicting interests.' (Lu et al., 2007, p.27). In their work, they designed a collaborative engineering process consisting of four phases that have to be addressed for an appropriate collaboration. Figure 3 depicts these four phases.

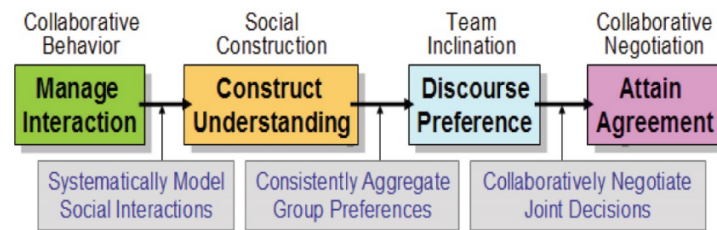


Figure 3. Engineering collaboration process by (Lu et al., 2007, p. 31).

The first phase presented in figure 3 is the management of social interactions. This includes the definition of stakeholders, goals, resources and workflows. The second phase is the construction of a common understanding. Herein, the goal understanding has to be synchronized for all stakeholders. Thirdly, a group preference has to be established instead of multiple individual stakeholder preferences. In the last phase, the team agreement has to be attained, which involves relevant negotiation activities (Lu et al., 2007, pp. 31ff.).

Stark and Stöckert (2009) have investigated another research approach in collaborative engineering. They examined errors created through virtual product creation (VPC), which denominates the usage of CAx for product development as computer-supported design activities (Albers and Lohmeyer, 2012). In the process of investigating errors, they defined interfaces in VPC that can be seen in figure 4, and discussed virtual collaborative engineering, also known as frontloading. As root causes for errors on the interfaces depicted in figure 4 Stark and Stöckert (2009) pointed out four issues:

- Domain-specific issues
- Technical issues
- Organizational issues
- Behavioral issues

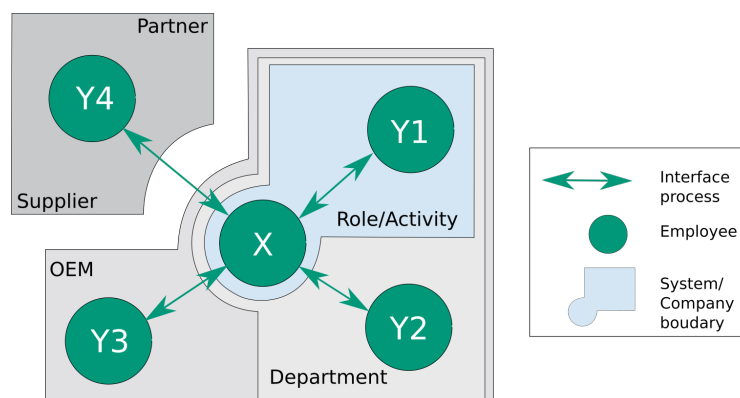


Figure 4. Interfaces of collaboration presented by Stark and Stöckert (2009).

The domain-specific issues mainly focus on the kind of project that can be defined for example by project size and depth of subcontracting. The technical issues refer to IT-problems and organizational issues for example on the processes. Behavioral issues refer for example to role and motivation conflicts.

Considerations for collaboration in MBSE

Based on the state of the art, multiple challenges for collaboration in general and in SE in particular could be identified. Table I gives an overview of the identified challenges and adds the reference that mentioned these challenge.

	1	2	3
C1 shared language/vocabulary	•	•	•
C2 management of interactions	•	•	
C3 exposition of data	•		•
C4 consistency of data	•	•	
C5 large collaboration networks	•		•
C6 system compatibility			•

Table I. Overview of identified challenges with • highlighting the sources of the challenge. References: 1 - Wouters et al. (2017a), 2 - Lu et al. (2007) and 3 - Stark and Stöckert (2009).

What can be seen in table I is that most challenges have been addressed by multiple researchers. As the column '1' referencing Wouters et al. (2017b) is explicitly focusing on MBSE the further references on the same challenge show that this seems to be a common challenge in collaboration and not only due to MBSE. In the following sections, these challenges shall be addressed with regard to MBSE and possible solutions shall be pointed out. It shall be mentioned that all considerations are meant as starting points for research approaches and have to be refined and tested in further research.

Used example

To depict the challenges in a demonstrative way the development of a Mars rover shall be used. As a Mars rover is a complex product in whose development multiple domains have to be integrated, this an appropriate example for the application of MBSE in a collaborative environment. The prostep ivip e.V. developed a SysML-model for the purpose of Cross-Discipline Lifecycle Collaboration (CDLC) (prostep ivip e.V., 2019), which is licensed under Apache license version 2.0 and therefore perfectly suitable for the current purpose. It is currently still under development by prostep ivip e.V. and thus neither fully translated nor complete.

Considering the process illustrated in figure 2 the product has to be defined in the first step. This can be compared to the second phase of collaborative engineering

process of Lu et al. (2007), labeled as 'construction of a common understanding' in figure 3. The SysML-model of prostep ivip e.V. uses a block definition diagram (bdd) as context diagram for this step, which can exemplarily be seen in figure 5.

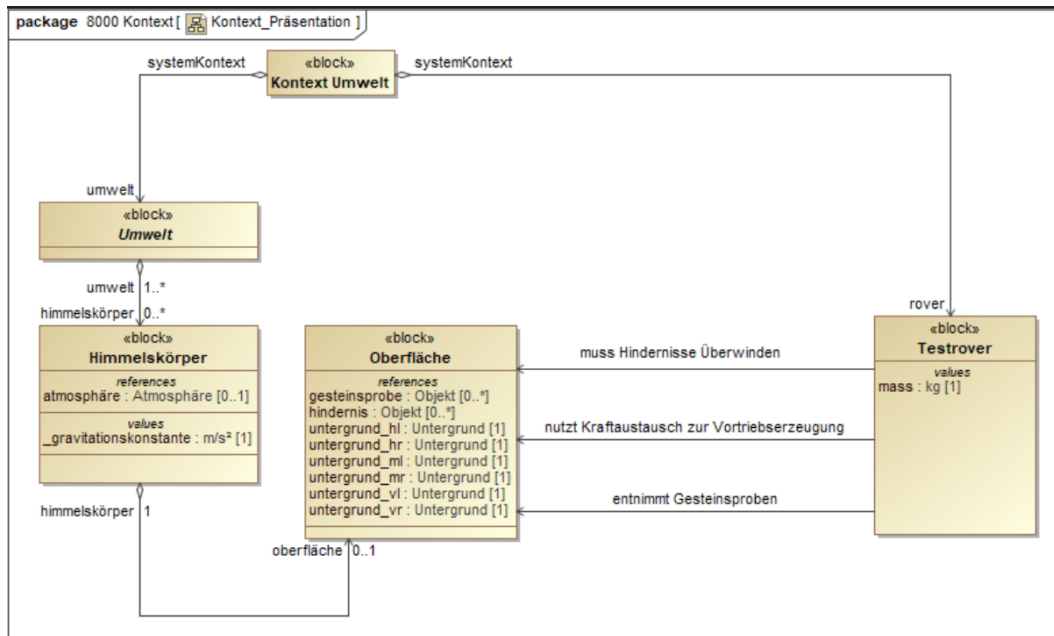


Figure 5. Mars rover Context Diagram.

This figure depicts the context in which the Mars rover shall be used in his operational time. It shows different objects it has to interact with as well as possible interaction with one of these.

In this figure, the first challenge (C1) is addressed in two ways: SysML as a graphical modeling language in MBSE can be used as a shared language between the various domains. The usage of models in this phase additionally supports the consistency of data (C4). However, SysML as well as other graphical modeling languages in that area still have some lack in understandability. This is at first the graphical modeling language itself that has to be learned by the involved stakeholders and the written language in form of vocabulary, which is used inside the various blocks. As mentioned earlier, the SysML-example of prostep ivip e.V. is currently developed in German and might not be understood by foreign stakeholders. This can be especially challenging considering a large collaboration network (C5). Here, the usage of different terms for the same element or the usage of the same term for different elements might be problematic, considering different language and domain languages. Wouters et al. (2017b) tried to handle this problem by semantic transfer of modeled information.

To manage the network of collaborations, a stakeholder overview can be modeled as well. This is presented for the SysML-model of the Mars rover in figure 6.

The shown diagram brakes down all stakeholders as actors from single roles to complete departments. When interconnected with further artifacts, the stakeholder

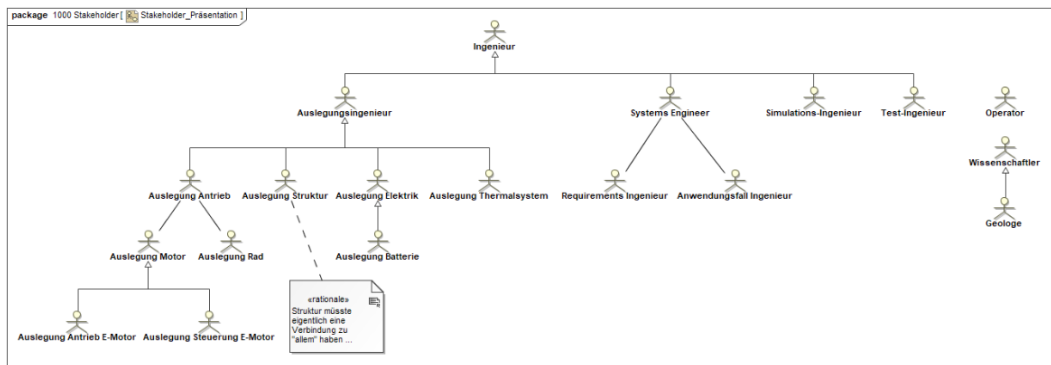


Figure 6. Mars rover Stakeholder overview as block definition diagram (bdd).

management (**C2** as well as **C5**) can be supported by MBSE and system model elements.

When going further in the process of figure 2, in the subsystem level and at the bottom of the 'V' the different stakeholders model their specific elements of the system which are required to build the complete product. This can be any models used in model-based engineering, may it be models of the control unit of the rover, SysML models of the subsystem architecture or CAD models of its physical parts. With respect to the large collaboration network of **C5** with every stakeholder in the collaboration using its own systems and its own internal knowledge, the exposition of data (**C3**) and the system compatibility (**C6**) become crucial factors. Eckl et al. (2015) have already addressed the system compatibility as they mentioned standard interfaces like OSLC for the exchange of information. This enables additional CSCW practices as for example the usage of web service and thus supports collaborative system development. The exposition of data (**C3**) can be controlled by combining the system modeling approaches of Stark and Schulze (2010) presented in figure 1. When the system elements relevant for intellectual property are linked only via their results (third form), only the relevant elements would be shared. This can be combined with the currently as ISO 10303-243 developed standard for Modelling and Simulation information in a collaborative Systems Engineering Context (MoSSEC) which can be found in more detail in Murton and Pollari. In such way, all relevant data can be shared without exposing confidential intellectual property. This is equal to delivering a component of the Mars rover with an appropriate user manual that list boundary conditions of usage but does not expose the inner functionality of the component.

The right side of the V-Model process for the development of the Mars rover faces the same issues as presented on the left and bottom part of the process.

In total, the six challenges of table I could be confirmed as well potentially solved on the example of the Mars rover model. In the following, they shall be summed up.

(C1) Shared language/vocabulary: The first challenge has been tackled in the Mars rover example by using a common graphical modeling language for the data exchange. By using this graphical modeling language, there occur the new problems of learning the modeling language as well as understanding the used vocabulary in sense of foreign language. This is confirmed by Königs (2012), who pointed out, that a broad knowledge of the language is necessary but has for example not been available to many employees in the automotive industry. Domain- and company-dependent vocabulary might be challenging as well. Königs (2012) for example mentioned that the high modifiability of UML and SysML leaves much room for interpretation and thus it might be even more difficult to understand the meaning of the diagrams. Transferring this to different languages and models it is hard, to reach a fully shared vocabulary. One approach has been provided by Wouters et al. (2017b) as they transformed the semantic information from one domain to another. Additionally, there exists much research on ontology, which could be used to reach this semantic consistency. For further reference, see Robinson and Bannon, Aslan et al. (2011) or Graves (2012).

(C2) Management of interactions: The orchestration is handled in the Mars rover example with a stakeholder diagram in SysML. It breaks down the different stakeholders and has to be interconnected to role and workflow concepts of other tools to regulate the interaction management defined by Lu et al. (2007) and Wouters et al. (2017b).

(C3) Exposition of data: This challenge becomes crucial when the stakeholders begin to include their knowledge into the system development, which is mainly the case in the subsystem and domain level. Apart from the method of transferring a reduced model to collaboration partner, which breaks with the philosophy of a system model as Single Source of Truth (SSoT), the modeling of the system model can be handled in the third form of Stark and Schulze (2010) by linking the results of the system elements. In that way, the intellectual property can be kept and the system model stays consistent. ISO 1030-243 will bring additional approaches by giving a standard for exchanging collaboration data with these models (Murton and Pollari).

(C4) Consistency of data: the system model as SSoT directly addresses the consistency of data. Research on and industrial implementations of new traceability approaches allow even better suspecting of inconsistencies and changes. While most of the time, the software solutions for the modeling with graphical modeling languages like IBM Rhapsody have their own tool-set for traceability integrated, there is some work on semantic web technologies, for example Wang et al. (2018), who addressed change propagation analysis making use of Web Ontology Language (OWL). This allows even more collaborative approaches, as the stakeholders are not bound to use the specific tools.

(C5) Large collaboration networks: The large collaboration networks mainly influence all other challenges, because the network multiplies the challenges of a single collaboration with the additional collaborations of the network. This challenge might not be addressed on its own, but shows that the solving of one of these challenges may influence further collaborations positively.

(C6) System compatibility: The system compatibility is an essential aspect in the subsystem and domain-specific phases. But even in the top-level phases it might not be possible for everyone to use the same tool to actively model into the system model. Standard formats (e.g. Requirements Interchange Format (ReqIF) for requirements) and standard interfaces (e.g. OSLC) are required to allow the participation of all stakeholders and are currently supported by a rising number of tool vendors.

Possible approach for handling current problems in collaborative MBSE

In this section, the challenges shall be fully addressed in one conceptual approach, which can be used for further investigation.

At first, a stakeholder overview is modeled in any desired modeling language. This stakeholder overview is used to design workflows and manage access rights of the various stakeholders. It shall be stored in a model repository, which is accessible for all stakeholders, such that they can view the defined interaction rules and can request or suggests changes easily through the model. To avoid incorrect modifications, the model should be made read-only to most roles in the collaboration network changes should be requested through a change management or issue tracking system. Only the interactions of the stakeholders inside the main collaboration should be managed inside this overview. Further collaborations of a collaboration network should be kept internal to not expose data that is not required for the collaboration. (C2, C3, C4 and C5)

In this model repository, a data model for ontology as well as the top-level requirements, functions and architecture elements should be stored, so that every participant of the collaboration can link the partial system elements to them. In that way, the shared system model takes a central role and can be send to communicate and manage all relevant information for a collaborative system development. The ontology data model for example can be extended with own definitions of all stakeholders such that the available definitions can be used, to translate domain- or company-dependent terms automatically. Wouters et al. (2017b) approach might be usable based on the input of this data model and thus support the shared vocabulary (C1 and C4).

When modeling further in the development processes the artifacts should be included into the system model, which does not imply the usage of a meta-model but the linkage with an approach of Stark and Schulze (2010) presented in figure 1. The systems architecture as well as functional architecture may be modeled with SysMT and further artifacts, e.g. the behavior of a systems element, modeled in

some other tool like MATLAB and only the results are then linked via OSLC. This approach supports the consistency of the system model while enabling a great system compatibility as well as avoiding an exposure of intellectual property (C3, C4 and C6).

SysMT of Königs (2012) is mentioned here instead of SysML due to its capability to display thumbnails with additional information as representations of the various elements. In the currently developed SysML v2 3D representations shall be included as well (Weilkiens, 2019), which will support the comprehensibility for all stakeholders drastically. This should be an important factor when choosing a modeling language, as 2D and 3D visual representations are most of the time more self-explanatory than blocks with text as used in most modeling languages, and thus support the common understanding (C1).

In general, as further system elements are created they should be added to the system model, by linking them to their relevant system elements. OSLC might be a good approach, as multiple tools, like IBM Rational or PTC Windchill already adapt to this standard. When this standard is used not just to link the tool suits directly to each other, but to link the artifacts to a system model with the approaches of Stark and Schulze (2010), the model can be used as consistent system data container comprising all relevant information for the development of the system of interest (C4 and C6). Figure 7 depicts this integration of all modeling parts in the system model via OSLC links.

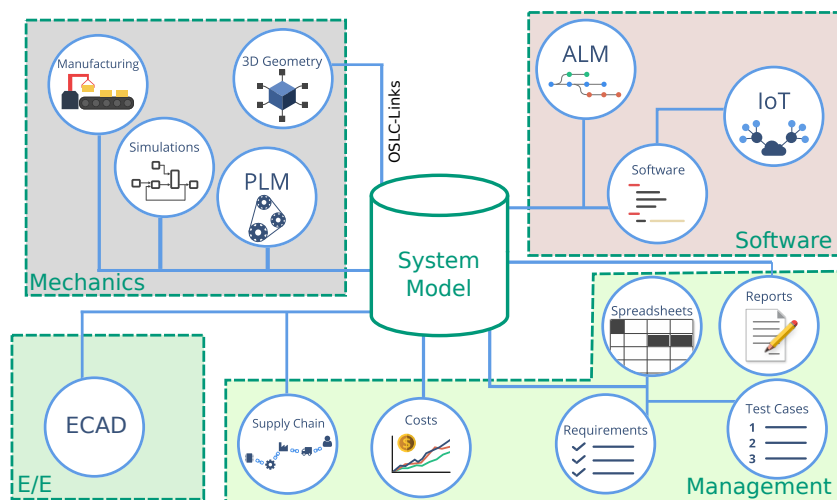


Figure 7. OSLC usage for MBSE adapted from OASIS (2019).

The depicted system model in the middle of figure 7 has not to be a meta-model, but may be any form presented by Stark and Schulze (2010) and shown in figure 1. The model elements of the different data sources around the system model, that are exemplary linked via OSLC-links to it, are connected to other system elements of the system model and thus include themselves inside of the overall system model. The linked data can then be used in any required form, e.g. as input for calculations and simulations or requirement refinement, which enables

current CSCW techniques as web services and Human-Computer-Interaction (HCI). To visualize this information in the best way, the combination of 3D objects and the accessible system information shall be investigated for feasibility in upcoming research.

With this MBSE-based approach a collaborative system development can be supported and enable further CSCW potentials. The system model itself can be seen as computer-supported means as well, as it requires computer-technology for proper functionality in this scenario. Thus, this approach can be seen as a computer-supported approach for collaborative systems development.

Conclusion and Outlook

The paper was intended to investigate collaboration in engineering and the potentials of MBSE as a computer-supported approach for collaborative systems development. To address this topic at a collaboration community like the European Conference on Computer-Supported Cooperative Work (ECSCW), an overview about MBSE as a transdisciplinary approach to develop complex systems in general has been presented followed by the state of the art for collaboration in cooperation in both MBSE and CSCW research.

The insights of the state of the art have been used to point out general challenges in collaboration for systems development. These challenges have been projected onto the example of a Mars rover's development.

Based on that, a development approach has been presented, that uses insights of the state of the art to address these challenges. The key element of the concept is the usage of the system model as central element of the systems development and the combination of the three approaches of system modeling presented by Stark and Schulze (2010) and shown in figure 1. With this MBSE-based approach, collaborative system development can be supported and the application of CSCW techniques can be enabled. Thus, it can be seen as computer-supported approach for collaborative system development.

In further research, this approach shall be prototypical implemented as well as tested and analyzed based on the mentioned challenges by combining the various approaches of system modeling and interconnecting different system infrastructures. 3D visualization shall be investigated in its applicability as well, as it is seen as a mean for better communication.

Acknowledgments

The authors want to acknowledge prostep ivip for licensing their Mars rover model under Apache license version 2 and thus allowing research to improve and use it properly.

References

- Albers, A. and Q. Lohmeyer (2012): ‘Advanced systems engineering - Towards a model-based and human-centered methodology’. In: I. Horvárt (ed.): *Tools and methods of competitive engineering: Proceedings of the ninth International Symposium on Tools and Methods of Competitive Engineering (TMCE 2012), Karlsruhe, Germany, May 7-11, 2012*. Ed.: I. Horváth. Karlsruhe, Germany, pp. 407–416.
- Aslan, K., N. Alhadad, H. Skaf-Molli, and P. Molli (2011): ‘SCHO: An and Ontology Based and Model for Com and puting Divergence and Awareness in Distributed and Collaborative Systems’. In: *ECSCW 2011: Proceedings of the 12th European Conference on Computer Supported Cooperative Work, 24-28 September 2011, Aarhus, Denmark, Springer*.
- Auricht, M., B. Beckmann-Dobrev, and R. Stark (2012): ‘An interdisciplinary approach to validate mechatronic systems in early product development stages’. In: *NordDesign 2012*. p. 8.
- Beier, G., A. Figge, U. Rothenburg, and S. Marwedel (2014): ‘A MBSE APPROACH FOR THE DEVELOPMENT OF COMPLEX TECHNICAL SYSTEMS’. *Test*.
- Buchholz, C., M. Tiemann, and R. Stark (2018): ‘Durchgängiges Prototyping mechatronischer Systeme im MBSE Entwicklungsprozess’. In: *Symposium Design fox X 2018*. pp. 128–140.
- De Carvalho, A. F. P., S. Hoffmann, D. Abele, M. Schweitzer, P. Tolmie, D. Randall, and V. Wulf (2018): ‘Of Embodied Action and Sensors: Knowledge and Expertise Sharing in Industrial Set-Up’. *Computer Supported Cooperative Work (CSCW)*, vol. 27, no. 3-6, pp. 875–916.
- Eckl, C., M. Brandstätter, and J. Stjepandić (2015): ‘Using the "Model-based Systems Engineering" Technique for Multidisciplinary System Development’. *Advances in Transdisciplinary Engineering*, vol. 2, no. Transdisciplinary Lifecycle Analysis of Systems, pp. 100–109.
- Estefan, J. A. (2008): ‘Survey of Model-Based and Systems Engineering and (MBSE) and Methodologies’. Technical Report INCOSE-TD-2007-003-02, International Council on Systems Engineering (INCOSE), Seattle, WA, USA.
- Friedenthal, S., A. Moore, and R. Steiner (2012): *A practical guide to SysML: the systems modeling language*. Waltham, MA: Morgan Kaufmann, 2nd ed edition. OCLC: ocn754518532.
- Graves, H. (2012): ‘Integrating Reasoning with SysML’. *INCOSE International Symposium*, vol. 22, no. 1, pp. 2228–2242.
- Haskins, C. (2011): ‘A historical perspective of MBSE with a view to the future’. vol. 21, no. 1, pp. 493–509.
- Hoffmann, S., A. F. P. de Carvalho, D. Abele, M. Schweitzer, P. Tolmie, and V. Wulf (2019): ‘Cyber-Physical Systems for Knowledge and Expertise Sharing in Manufacturing Contexts: Towards a Model Enabling Design’. *Computer Supported Cooperative Work (CSCW)*, vol. 28, no. 3-4, pp. 469–509.
- Huldt, T. and I. Stenius (2018): ‘State-of-practice survey of model-based systems engineering’. *Systems Engineering*, vol. 22, no. 2, pp. 134–145.
- INCOSE (2015): *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Wiley, 4 edition.
- INCOSE (2019): ‘About Systems Engineering’. <https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition>. Accessed on 25th January 2020.

- INCOSE Technical Operations (2007): *Systems Engineering Vision 2020*, No. INCOSE-TP-2004-004-02. Seattle, WA: International Council on Systems Engineering, version 2.03 edition.
- ISO 15288 (2015): ‘Systems and software engineering — System life cycle processes’. Technical report, International Organization for Standardization.
- ISO 29110 (2016): ‘Systems and software engineering — Lifecycle profiles for very small entities (VSEs)’. Technical report, International Organization for Standardization.
- Jacobs, S., N. Wengrowicz, and D. Dori (2014): ‘Defining Object-Process Methodology in Web Ontology Language for Semantic Mediation’. In: *2014 IEEE International Conference on Software Science, Technology and Engineering*. IEEE.
- Jia, L. and H. Zhang (2007): ‘Supporting Collaborative Product Design with a Web Service and HLA Based Architecture’. In: *2007 11th International Conference on Computer Supported Cooperative Work in Design*. IEEE.
- Kim, W. and M.-J. Chung (2005): ‘Collaboration in design and manufacturing process using Web services semantics’. In: *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, 2005*. IEEE.
- Königs, S. F. (2012): ‘Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung’. Ph.D. Thesis, TU Berlin.
- Königs, S. F., G. Beier, A. Figge, and R. Stark (2012): ‘Traceability in Systems Engineering: Review of industrial practices, state-of-the-art technologies and new research solutions’. *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 924–940.
- Lindow, K., P. Müller, and R. Stark (2011): ‘NEW JOB ROLES IN GLOBAL ENGINEERING – FROM EDUCATION TO INDUSTRIAL DEPLOYMENT’. In: *Culley, Hicks et al. (Hg.) 2011 – ICED 11*. pp. 374–385.
- Lu, S.-Y., W. Elmaraghy, G. Schuh, and R. Wilhelm (2007): ‘A SCIENTIFIC FOUNDATION OF COLLABORATIVE ENGINEERING’. *CIRP Annals*, vol. 56, no. 2, pp. 605–634.
- Murton, A. and G. Pollari. ‘ISO 10303 (STEP) MoSSEC edition 1 - Application Protocol For Modelling and Simulation information in the collaborative Systems Engineering Context (MoSSEC)’.
- OASIS (2019): ‘Open Services for Lifecycle Collaboration’. <https://open-services.net/>. Accessed on 16th of January 2020.
- prostep ivip e.V. (2019): ‘Cross-Discipline Lifecycle Collaboration’. <https://www.prostep.org/en/projects/cross-discipline-lifecycle-collaboration/> Accessed on 16th April 2020.
- Robinson, M. and L. Bannon, ‘Questioning Representations’. In: *Proceedings of the Second European Conference on Computer-Supported Cooperative Work*.
- Roques, P. (2016): ‘MBSE with the ARCADIA Method and the Capella Tool’. In: *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, Toulouse, France.
- Schäfer, W. and H. Wehrheim (2010): ‘Model-Driven Development with Mechatronic UML’. In: *Graph transformations and model-driven engineering : essays dedicated to Manfred Nagl on the occasion of his 65th birthday*. Springer Berlin Heidelberg, pp. 533–554.

- Stark, R. and E.-E. Schulze (2010): 'Need and potentials to improve Systems Engineering by future PLM Solutions'.
- Stark, R. and H. Stöckert (2009): 'Interface errors - an issue in complex virtual product creation'. In: M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad, and U. Lindemann (eds.): *Proceedings of ICED'09*. Glasgow, pp. 465–476, Design Society.
- Sünnetcioglu, A., P. Mochine, O. von Dungern, and R. Stark (2017): 'SysML-Modelle maschinell verstehen und verknüpfen'. In: GfSE (ed.): *Tag des Systems Engineering 2017*. Carl Hanser Verlag GmbH & Co. KG, pp. 157–166.
- Tomiyama, T., E. Lutters, R. Stark, and M. Abramovici (2019): 'Development capabilities for smart products'. *CIRP Annals*, no. 68, pp. 727–750.
- VDI 2206 (2004): 'Design methodology for mechatronic systems'. Guidance, VDI-Verlag GmbH.
- Wang, H., V. Thomson, and C. Tang (2018): 'Change propagation analysis for system modeling using Semantic Web technology'. *Advanced Engineering Informatics*, vol. 35, pp. 17–29.
- Wang, J. X. and M. X. Tang (2005): 'Knowledge representation in an agent-based collaborative product design environment'. In: *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, 2005*. IEEE.
- Weilkiens, T. (2019): 'SysML v2 - The Next Generation'. p. 25, HOOD Group.
- Wouters, L., S. Creff, E. E. Bella, and A. Koudri (2017a): 'Collaborative systems engineering: Issues & challenges'. In: *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE.
- Wouters, L., S. Creff, E. E. Bella, and A. Koudri (2017b): 'Towards Semantic-Aware Collaborations in Systems Engineering'. In: *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE.