



Combining and evaluating function-oriented solutions in model-based systems engineering

Lukas Irnich¹ · Georg Jacobs¹ · Thilo Zerwas¹ · Christian Konrad¹

Received: 9 December 2022 / Accepted: 19 January 2023
© The Author(s) 2023

Abstract

Increasing complexity and shorter innovation cycles require a rethink in the development of mechatronic products. A function-oriented development process using model-based systems engineering (MBSE) methods promises to identify technical solutions via solution-neutral functional architectures. Since each function can typically be realized by multiple, alternative solutions the combinatorial number of overall system solutions explodes. Existing MBSE approaches enable to establish functional architectures as well as support the verification of solutions. However, evaluating different solution combinations with the current state of research involves repetitive and time-consuming modeling which leads to a heuristic selection of potentially suboptimal solutions. In this paper, we present a method for the systematic and function-oriented composition and model-based evaluation of solution combinations. Alternative solutions are structured based on a functional architecture, so that each possible solution combination can be composed and simulated with a minimum of effort, considering physical interactions and the efficiency. Based on simulation results, solution combinations can be verified against requirements and evaluated striving for the best solution combination. The elaborated method extends an existing MBSE approach and is illustrated by the running example of an electrical coolant pump of a passenger vehicle.

Kombination und Evaluation funktionsorientierter Lösungen in der modellbasierten Systementwicklung

Zusammenfassung

Steigende Komplexität und kürzere Innovationszyklen erfordern ein Umdenken bei der Entwicklung mechatronischer Produkte. Ein funktionsorientierter Entwicklungsprozess mit Methoden des Model Based Systems Engineering (MBSE) verspricht, technische Lösungen über lösungsneutrale Funktionsarchitekturen zu identifizieren. Da jede Funktion typischerweise durch mehrere, alternative Lösungen realisiert werden kann, explodiert die kombinatorische Anzahl möglicher Gesamtsystemlösungen. Bestehende MBSE-Ansätze unterstützen sowohl die Erstellung funktionaler Architekturen als auch die Verifikation von Gesamtsystemlösungen. Die Evaluierung verschiedener Lösungskombinationen erfordert nach dem derzeitigen Stand der Forschung jedoch eine wiederholte und zeitaufwändige Modellierung, die zu einer heuristischen Auswahl potentiell suboptimaler Lösungen führen kann. In diesem Beitrag wird eine Methode zur systematischen und funktionsorientierten Erstellung und modellbasierten Bewertung von Lösungskombinationen vorgestellt. Alternative Lösungen werden auf Basis einer funktionalen Architektur strukturiert, so dass jede mögliche Lösungskombination unter Berücksichtigung physikalischer Wechselwirkungen und der Effizienz mit minimalem Aufwand gebildet und simuliert werden kann. Basierend auf den Simulationsergebnissen können Lösungskombinationen anhand der Anforderungen verifiziert und bewertet werden, um die beste Lösungskombination zu finden. Die erarbeitete Methode erweitert einen bestehenden MBSE-Ansatz und wird am Beispiel einer elektrischen Kühlmittelpumpe erläutert.

✉ Lukas Irnich
lukas.irnich@imse.rwth-aachen.de

¹ Institute for Machine Elements and Systems
Engineering, RWTH Aachen University,
Eilfschornsteinstraße 18, 52062 Aachen, Germany

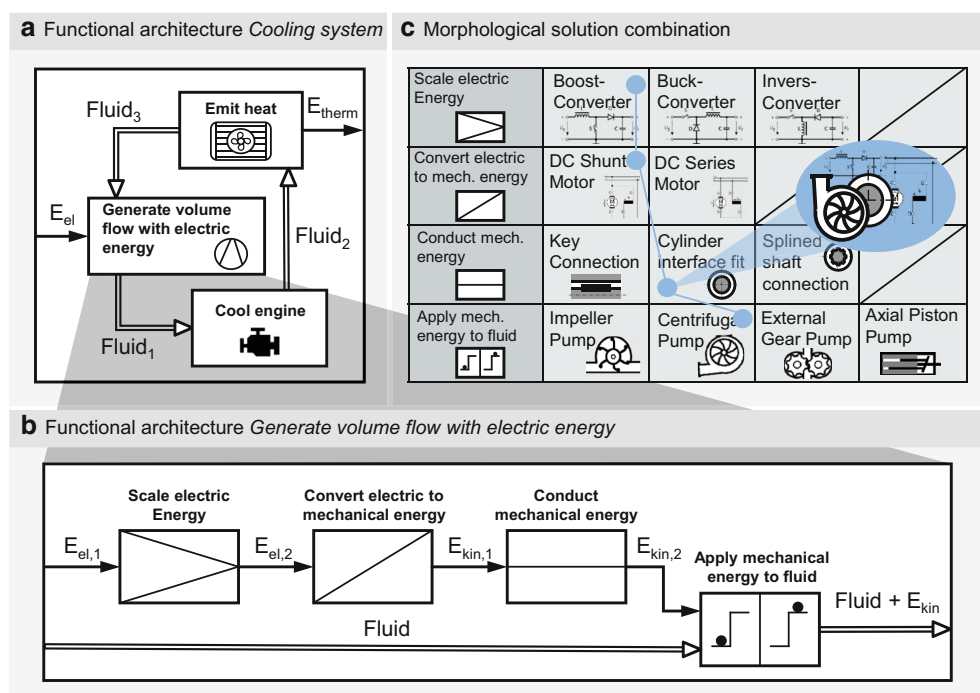
1 Introduction

Increasing complexity and shorter innovation cycles represent a major challenge for the development of mechatronic products. This is also evident in the automotive sector, where vehicles must be developed from interacting subsystems of mechanical, electrical and software engineering in a higher quality, shorter time and at lower cost [1]. One approach for cross-domain development is function-oriented product development. The key idea is to develop a functional architecture that represents the functional behavior of the product in a solution- and domain-neutral way [2]. Each sub-function of the functional architecture can typically be realized by several, alternative, technical solutions [3, 4], which are assigned to the respective sub-functions. Based on these functionally structured solutions, overall system solutions can be combined by selecting a technical solution for each sub-function. This combinatorics results in a very large number of possible overall system solutions via the potency of the amount of solutions and functions. This phenomenon is also known as “combinatorial explosion” [27]. This results in the central challenge of determining the best overall system solution for a specific development project among a large number of possible overall system solutions.

An example of a mechatronic product that can be used to illustrate the combinatorial explosion of overall system solutions is the electric coolant pump of a passenger car, which will be used as a running example in this paper based on [2, 5]. The purpose of the coolant pump is to deliver coolant in a cooling circuit for temperature control of the drive train and the cabin (Fig. 1a). The functional archi-

tecture for generating the volume flow describes that an incoming electrical power is scaled and then converted into a mechanical power. This mechanical power is conducted and finally applied to the coolant for its pumping (Fig. 1b). For each sub-function, there are several possible technical solutions. For example, the sub-function *Apply mechanical Energy to Fluid* can be implemented as a *Centrifugal Pump*, *Impeller Pump*, *External Gear Pump* or *Axial Piston Pump* (Fig. 1c). By combining one technical solution per sub-function, a pump can be synthesized, which for example consists of a *Boost converter*, a *DC shunt motor*, a *Key connection* and a *Centrifugal pump* (Fig. 1c). In total, 72 different solution combinations can be formed in the example shown. However, typical development scopes include more functions, for which also more technical solutions are known. Therefore, in reality, thousands of different possible solution combinations are often available, all of which would need to be evaluated during the product development process, to find the most promising ones. This challenge is further increased by the parameterization of the solution combinations. Only when concrete parameters for geometrical or material characteristics of the technical solutions are available, calculations can be executed to provide valid reasons for the selection or exclusion of a solution combination. Without parameterization and calculations, the selection or exclusion is based on subjective and possibly incorrect assumptions. If we assume that only five discrete parameterizations are available for each technical solution, the 72 solution combinations shown above increase dramatically to 45,000 parameterized solution combinations. In reality, the number of possible parameterizations per

Fig. 1 Functional architecture of the *Cooling system* (a), Functional architecture *Generate volume flow with electric energy* (b), Morphological solution combination (c)



technical solution is not limited to five. A parameterization represents the combination of different parameters and causes a combinatorial explosion by itself.

In document-based product development, a morphological box is typically used to represent alternative technical solutions. However, the composition of solution combinations in the morphological box remains a manual activity. Due to the lack of formalized representation of technical solutions in traditional document-based product development, solution combinations cannot be efficiently evaluated by simulation models, leading to heuristic and subjective evaluation processes and results. Also, the subjective evaluation is carried out manually. Because of the manual execution and heuristic and subjective results, document-based methods are not suitable for generating and evaluating a large number of solution combinations. This does not ensure that the entire solution space is considered or that an optimized solution combination is identified.

Model-based systems engineering (MBSE) [6] provides a promising approach for mastering the combinatorial explosion in function-oriented development. Meanwhile, approaches for modeling the functional architecture of a mechatronic product including the technical solutions in a system model are known, which can be verified with integrated simulation models [2, 5, 7]. Up to now, however, there has been no approach for modeling alternative technical solutions, combining them into overall system solutions and evaluating them.

Hence, the contribution of this paper is a method that allows

1. the function-oriented capturing of alternative technical solutions,
2. the systematic composition of solution combinations,
3. the simulation of the physical behavior of solution combinations without additional effort, and
4. the evaluation of solution combinations against predefined criteria.

The paper is structured as follows: In Sect. 2, the state of research regarding the contributions described above is presented. Based on that, in Sect. 3 the research question and the research hypotheses are derived. A method to systematically structure and model all alternative technical solutions based on a functional architecture is presented in Sect. 4.1. Section 4.2 shows how each possible solution combination can be systematically composed based on this structuring and modeling. The subsequent simulation of the physical behavior of the solution combinations without additional modeling efforts is presented in Sect. 4.3. The subsequent evaluation of the solution combinations for defined evaluation criteria follows in Sect. 4.4. and 5 finally summarizes the contributions of the paper.

2 State of research

In Sect. 1, the need to combine and evaluate technical solutions in a function-oriented manner for the cross-domain development of mechatronic products was described. The following criteria for analyzing the state of the research can be derived from this problem statement. First of all, in order to capture technical solutions in a function-oriented manner, it is necessary to model the functional architectures of mechatronic products (criteria 1). In addition, it must be possible to formalize alternative technical solutions in such a way that they can be allocated to sub-functions of a functional architecture (criteria 2). Furthermore, the combination process of solutions must be feasible with a minimum of modeling effort in order to keep up with the combinatorial explosion. Therefore, the method should only require initial effort to generate a model that can be used for generating solution combinations via the simple selection of individual predefined technical solutions (criteria 3). The same applies to the simulation of solution combinations, which must be possible without further significant modeling work (criteria 4). Finally, the model-based evaluation of solution combinations should initially be possible based on the simulation results (criteria 5).

In recent years, a number of methods have been published that enable the function-oriented development of mechatronic products in a model-based way. Following, a selection of these methods is evaluated according to the previously defined criteria. Weilkiens describes in [8] a pragmatic and user-oriented approach for modeling requirements and system architectures (also known as SYSMOD). This method allows the modeling of functional and solution architectures, but does not support the modeling process via predefined development artifacts such as functional flows. In principle, solutions can be modeled, but their simulation is not possible in an integrated manner [8]. An approach of a modeling framework for model-based development of embedded system elements (also known as SPES) is proposed by Pohl et al. in [9]. The utilized data model enables the representation of functional architectures and alternative logical elements as well as the verification of requirements by simulation methods. But since the logical elements are not structured function-oriented down to elementary level, the combination of solutions cannot completely follow the functional architecture resulting in manual and effortful combination procedures. A model-based development process of cybertronic products and production systems (also known as mecPro)² is proposed by Eigner et al. [10]. The process focuses on the data-consistent integration of all involved development disciplines. It is function-oriented and allows the simulation of the physical behavior on the “technical solution level”. However, the simulation models are not structured

Table 1 Method evaluation for model-based combination, simulation and verification of function-oriented solutions (range: ++, +, o, –, --)

Evaluation criteria	Weilkiens [8]	Pohl et al. [9]	Eigner et al. [10]	Kruse [11]	Moeser et al. [13, 14]	Jacobs et al. [2, 5, 7, 15, 16]
1. Modeling of functional architectures	+	++	++	++	++	++
2. Modeling of alternative function-oriented solutions	–	o	o	+	+	++
3. Composition of solution combinations	--	–	–	--	–	o
4. Simulation of solution combinations	--	–	–	--	--	o
5. Evaluation of solution combinations	--	–	–	--	--	–

in a functional way, resulting in additional effort for generating a simulation model for each solution combination [10]. Another approach by Kruse aims to reduce modeling efforts in multidisciplinary concept development by reusing SysML-based submodels with the help of a model library [11, 12]. In the described process sub-functions can be allocated to alternative solutions and simulation models, however the extent and purpose of allocated simulation models do not automatically match to those of the sub-solutions but, often higher-level and more extensive systems. The combination of reused models is done by manually copying and pasting them, which in turn requires extensive modeling efforts. Moeser et al. describe in [13, 14] a method (also known as FAS4M) that aims to represent the function-design-relationship of design-neutral block diagrams to specific component geometries. However, the method partially describes solutions via sketches or at least without a parametric formalization that allows the seamless integration of simulation models which enables the efficient evaluation of solution combinations [13, 14]. The approach of Jacobs et al. [2, 5, 7, 15–17] (also known as motego method) focusses on the seamless development of mechatronic products in a cross-domain, model-based and function-oriented manner. This method is characterized by the use of function-oriented solutions, which enable model-based simulation and evaluation of overall system solutions early in the development process via formalized physical effects and active surface sets [5, 7]. The approach allows the combination of solutions but requires high modeling efforts since it needs to be done manually. The same applies to the simulation of solution combinations which requires the modeling of separate workflows for each combination.

Overall, no approach could be identified in the state of research that meets all evaluation criteria (cf. Table 1). Therefore, the need for research is formulated in the following chapter.

3 Research question and hypotheses

For the cross-domain development of mechatronic systems, it is necessary to combine and evaluate a large number of alternative technical solutions in a function-oriented manner

(cf. Sect. 1). From this problem criteria were derived in Sect. 2, which are currently not yet been overcome by the state of research. Therefore, the following research question is formulated for this paper:

How can solution combinations be composed systematically and function-oriented as well as evaluated in a model-based manner?

To answer this question, four research hypotheses are formulated:

1. Alternative technical solutions can be structured using a SysML-based language via a functional architecture.
2. Solution combinations can be efficiently composed by inheriting system relationships.
3. The physical behavior of composed solution combinations can be initially parameterized and calculated via integrated simulation models.
4. Solution combinations can be verified and evaluated against requirements based on the simulated physical behavior.

Since the method developed in this paper only supports a specific section of the product development process, it will be developed as an extension of an existing method that fulfills most of the criteria. For this purpose, the preliminary work of Jacobs et al. [2, 5, 7, 15] was selected, which provides a method for the construction of functional architectures, the representation of individual technical solutions as well as their calculation with integrated simulation models. In the development of the method in this paper, particular value is placed on a connectivity to the language profile of [16], which makes the preliminary work with a SysML-based [18] language efficiently usable.

4 Combination and evaluation of technical solutions

In this chapter, a method is presented that allows the function-oriented combination and evaluation of technical solutions and thus focuses on the derived criteria in Sect. 2.

The method is divided into four parts, each is presented in a subchapter and addresses one hypothesis (cf. Sect. 3). In Sect. 4.1 we present a method for the function-oriented modeling of alternative technical solutions. Sect. 4.2 shows the systematic composition of solution combinations. Sect. 4.3 shows how composed solution combinations can be simulated initially. Sect. 4.4 shows how simulated solution combinations can be verified and evaluated against requirements without further modeling efforts.

4.1 Creation of a solution framework model

In order to be able to capture alternative technical solutions in a function-oriented manner, a functional architecture is required. For the subsequent evaluation of the solutions requirements are needed too. Therefore, requirements and functions are modeled according to [2, 5, 7, 15] as the basis of the proposed method.

Modeling requirements and functions According to [2, 5], requirements can be differentiated into design requirements and functional requirements. Using the example of an electric coolant pump, design requirements are set for the operating point of the pump based on parameters commonly used on the market [28]. For example, a power supply with a voltage of 12.5 V has to be used. A volume flow rate of 360 L/min should be pumped while 0.8 bar pressurize the pumps inlet and 1.2 bar the outlet. A maximum electrical power of 2000 W can be provided for the pump operation by the alternator. Furthermore, the design requirement of cavitation-free operation must be met (Fig. 2a).

According to [2, 5], functional architectures consist of sub-functional architectures and elementary functions that are not subdivided further. In addition to the hierarchical structuring, the functions are linked horizontally via functional flows in order to represent physical interactions with regard to exchange of material, energy and signals [3]. In the example of the electric coolant pump, the overall function is the generation of a volume flow. Further, the pump must be operated using electrical energy. This overall function is here divided into subfunctions, which describe how the incoming electrical energy is scaled and converted into mechanical energy, which is ultimately applied to the coolant for its pumping (Fig. 2b).

Modeling abstract solutions According to [2, 5], following the requirements and functions, concrete technical solutions can be modelled, whose physical behavior can be calculated via integrated simulation models and workflows even before three-dimensional components and assemblies are designed [2, 7, 17]. With this modeling method, a separate workflow would have to be modeled for the simulation of each solution combination formed in Sect. 4.2. To reduce this modeling effort, the concept of abstract solutions is proposed here as an intermediate level instead of directly linking functions to concrete solutions. An abstract solution is a kind of placeholder that can be used to build workflows before solutions are combined. Later, this placeholder is consistently replaced by concrete technical solutions so that the predefined workflows can still be validly executed. This means that workflows do not have to be built for every solution combination, but only once.

A challenge lies in the modeled direction of the inherited functional flows. In [2, 5], functional flows are modeled di-

Fig. 2 Design requirements (a) and functional architecture (b) of the electric cooling pump

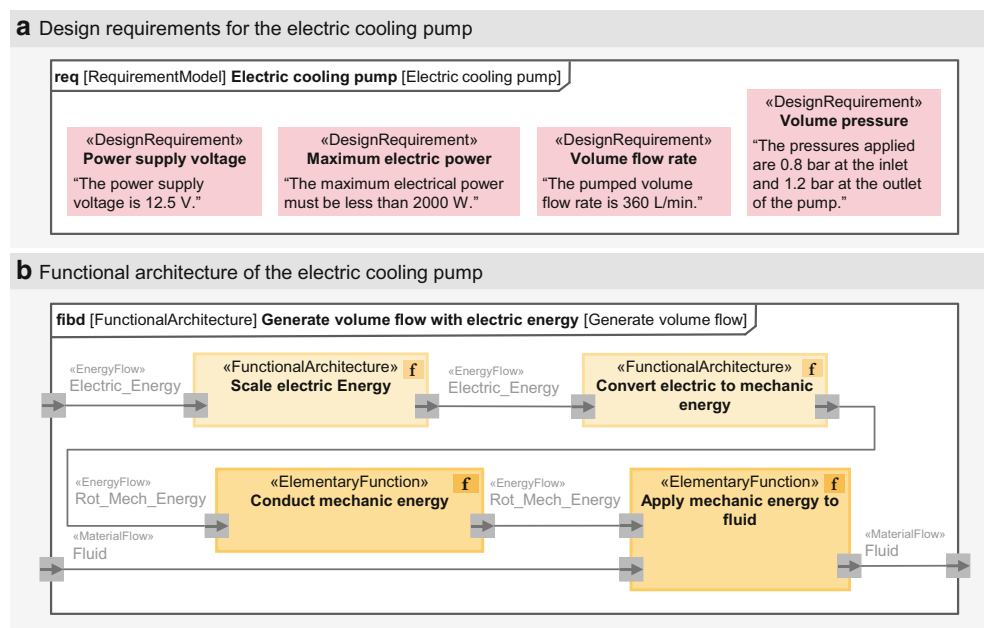
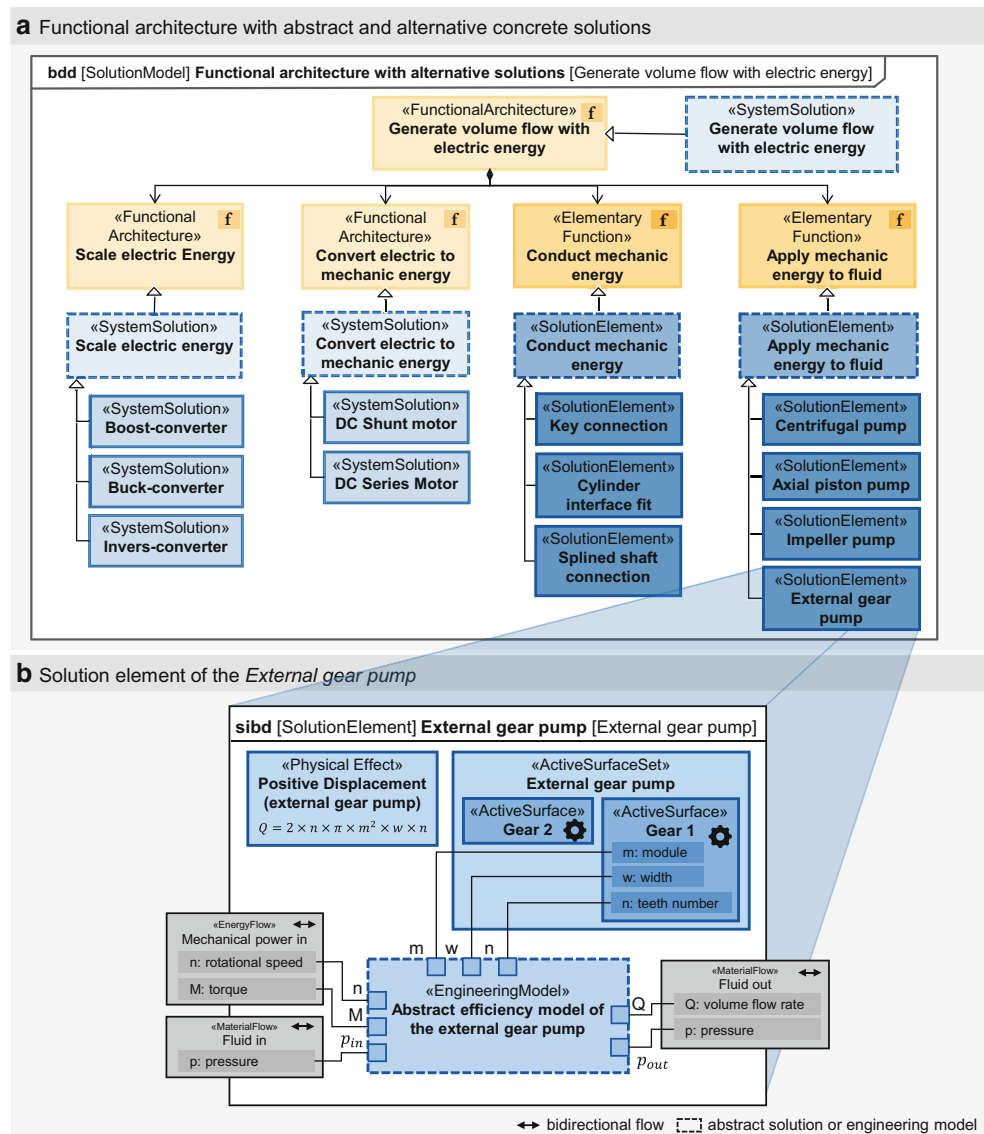


Fig. 3 Functional architecture with alternative solutions (a), solution element of the *External gear pump* (b)



rectionally to indicate in which direction e.g. a mechanical power flows through the functional architecture. In addition to the direction of flow intended by the engineer, however, mechanical systems in particular are characterized by physical interactions (also referred to as *actio* and *reactio*). In order to be able to calculate these physical interactions correctly, it may sometimes be necessary to calculate in the opposite direction of the functional flow according to [2, 3, 5]. Since the simulatability of the later formed solution combinations (cf. Sect. 4.3) is essential to master the combinatorial explosion in the function-oriented development, the abstract solutions are formalized with bidirectional ports.

Starting from the functional architecture, an abstract solution is modeled for each subfunction via an inheritance relation so that all properties are inherited (Fig. 3a). The directional flows inherited in this process are redefined as bidirectional flows for the reasons mentioned earlier. In or-

der to be able to form solution combinations in Sect. 4.2, an abstract overall system solution is also created at the level of the overall functional architecture via an inheritance relation (Fig. 3a). Within this abstract overall system solution, all preparations for the later simulation of solution combinations without additional modeling efforts (cf. Sect. 4.3) are made by redefining the inherited functions to abstract sub-solutions as well as linking them bidirectionally (Fig. 5).

Modeling alternative solutions In mechanical engineering, functions can often be realized by a multitude of technical solutions, which have to be considered in the development process [3]. These concrete technical solutions can be modeled as solution elements or system solutions [2]. Following the concept of principle solutions known from design methodology [1, 3, 4], solutions consist of a physical effect, an active surface set and were supplemented by domain

models and workflows by [2]. How a solution can physically achieve the desired functional behavior is defined by its physical effect. The active surface set describes the most important geometric and material parameters, which influence the physical effect and its behavior [2, 3, 5]. For example, a technical solution for pumping a volume flow is the *External gear pump*, which is based on the physical effect of positive displacement and is geometrically characterized by two interacting gears (Fig. 3a). Thus, concrete technical solutions can be modeled according to [2]. These concrete technical solutions are linked via an inheritance relation to the abstract solution of the sub-function they specify. Due to the generalization relationship used, the concrete solutions inherit all elements of the abstract solutions, including their bidirectional ports, which are essential for an efficient simulation.

Modeling 1d simulation models In addition to physical effects solutions contain all classified domain models that are relevant for testing and designing the solution [15, 19]. The evaluation of the solution combinations (cf. Sect. 4.4) will be exemplified in this paper by the efficiency, which is represented in 1d simulation models. The efficiency of an *External gear pump* for example can be calculated according to [20] via an experimentally determined characteristic curve of volumetric and frictional losses as functions of the pressure difference, displacement volume, density of the fluid and rotational speed. These functions can be implemented in a solution element together with the physical effect in a 1d simulation model either directly in a constraint

or via an interface in an external simulation tool (Fig. 3b). With this approach, sub system solutions can also be modeled as a black box in a first approximation instead of decomposing them in detail into solution elements. In this paper, the system solutions of the *Boost*, *Buck* and *Invers converter*, among others, were modeled using MATLAB Simulink simulation models [21].

Depending on the solution combination, the required inputs and outputs of a 1d simulation model may change, so that from a mathematical point of view, different transformations of the 1d simulation models are required. When using specific tools that enable MBSE in product development, such as CATIA Magic (formerly Cameo Systems Modeler) [22], mathematical formulas cannot be transformed automatically. Therefore, all possible transformations of the simulation models must be initially formed and captured as specifications of an abstract simulation model to reuse it for any possible input-output combination. For the 1d simulation model of the *External gear pump* three transformations are possible (Fig. 4). By inheriting the ports, all transformed variants of the simulation model can be included a priori and later specifically selected and executed without causing new modeling efforts. The abstract simulation model can be replaced by one of the specific transformations. This creates the basis for calculating the overall physical behavior regarding the efficiency of solution combinations in Sect. 4.3 without additional modeling effort.

The modeling of all possible transformations is initially more time consuming than the modeling of the actually

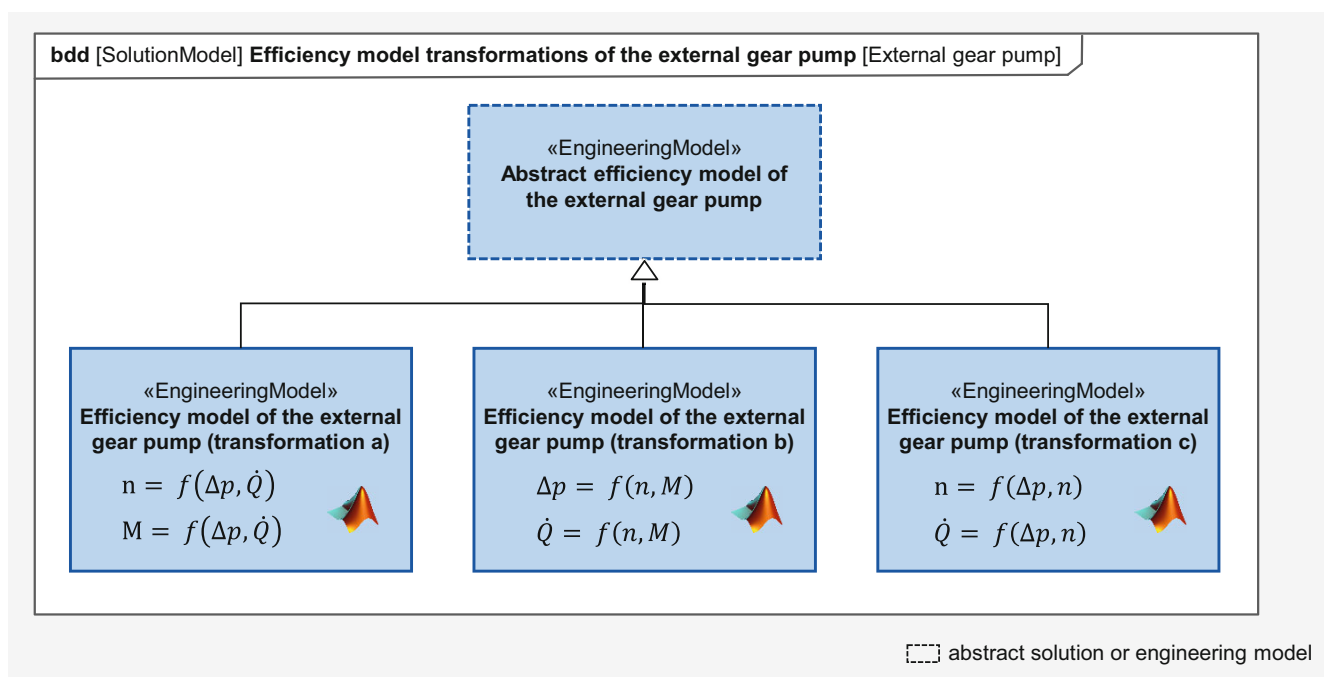


Fig. 4 Efficiency model transformations of the external gear pump

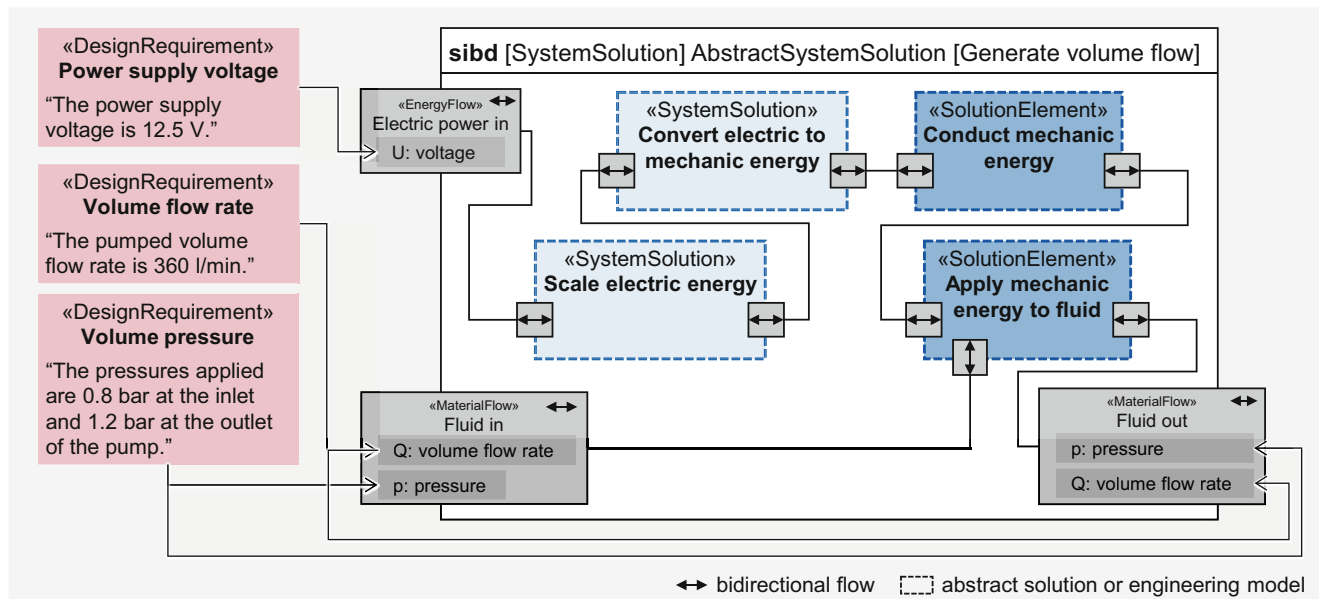


Fig. 5 Abstract system solution with linked design requirements

required ones. Without knowing the overall context, however, it is not yet possible to select a specific transformation. Furthermore, a long-term reuse of the solution element is only possible when all transformations have been formed. The increased modeling effort thus pays off through the reuse of the simulation model and the solution.

The evaluation of solution combinations with an efficiency can be implemented in solution models quite easily. For the calculation of the efficiency at the level of individual solutions, the required formulas and calculation models, as shown above for the external gear pump, are known and sufficiently described in the current state of research. However, the solution evaluation in product development depends on many criteria of multiple disciplines, such as costs, design space or weight. Here, model-based evaluation via 1d simulation models is especially difficult because the required models are missing according to the current state of research.

Modeling alternative parameter sets The calculation and evaluation of the physical behavior of a solution combination requires a concrete parameterization. Alternative technical solutions can differ qualitatively by other physical effects (e.g. positive displacement or centrifugal force) or active surface sets (e.g. external gears or piston and chamber), as well as quantitatively by active surface sets with different parameterizations (e.g. external gears with the width of 20 or 35 mm). In the function-oriented modeling shown so far, only solutions that differ qualitatively have been captured. This state is now extended in reference to different parameter sets for the active surface set for each solution. For this purpose, parameter sets with concrete parameter values for

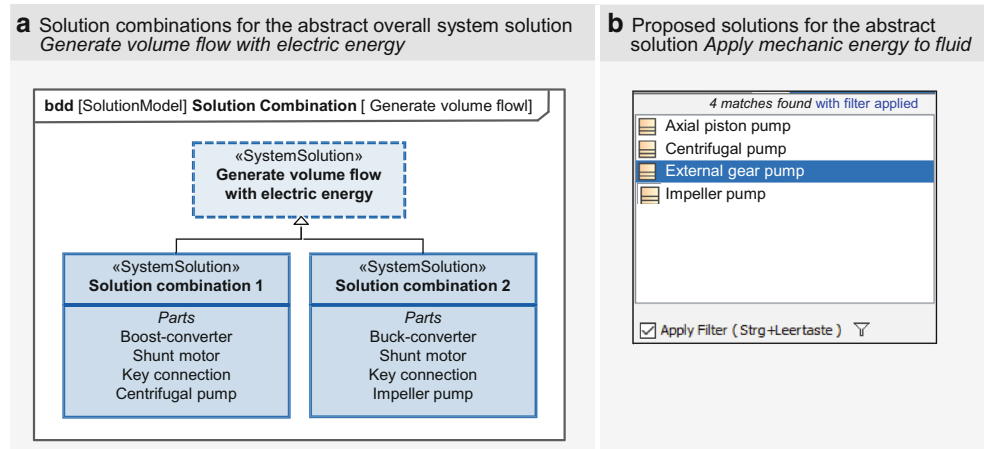
all value properties are modeled as instances of the active surface sets. For the electric coolant pump, the parameter sets were determined using exemplary products available on the market. Five parameter sets were modeled for each solution, representing a relatively wide range of sizes. In reality, far more parameter sets are possible. Considering the combinatorial explosion, the number is limited to five (cf. Sect. 1).

With the help of these parameter sets, a large number of parameterized solution combinations can be tested in an early development phase. Of course, the technical optimum can also lie between two parameter sets. Therefore, in the subsequent development, it is necessary to subsequently design and optimize the best identified solution combinations in more detail. This approach represents a compromise between a broad and objective consideration of all possible solutions and the subsequent, well-founded detailing of one or a few solution combinations. The number of parameter sets directly influences how precisely the optimum can be constrained, but also how large the combinatorial explosion turns out to be (cf. Sect. 1).

Modeling requirement linkages The composed solution combinations have to be verified against requirements without triggering additional modeling efforts. The linking of requirements can be accomplished in two different ways, depending on whether they refer to the entire solution combination or to a single solution element.

Design requirements [2], which refer to the entire solution combination, are linked to the abstract system solution, so that all generated solution combinations inherit this link directly. In the example of the electric coolant pump,

Fig. 6 Solution combinations for the abstract overall system solution *Generate volume flow with electric energy* (a) and proposed solutions for the abstract solution *Apply mechanical energy to fluid* while defining a new solution combination (b)



among other things, the requirement for the maximum allowed power has to be verified. For this purpose, the electrical current and voltage contained in the flow of electric power entering the overall system can be calculated during the simulation (cf. Sect. 4.3) and constantly compared with the limit values in the requirement. Therefore, this requirement is linked to the electric current and voltage of the functional flow of electric energy (Fig. 5).

Design requirements that only refer to individual solutions can be directly linked to the corresponding properties of this solution according to [2]. For example, cavitation-free operation is required for the electric coolant pump. This requirement refers exclusively to the alternative solutions which apply mechanical energy to the coolant. For example, for centrifugal pumps, the occurrence of cavitation can be predicted by exceeding a critical rotational speed [23, 24]. This critical rotational speed can be calculated based on the physical boundary conditions and the active surface sets according to [24]. Therefore, this mathematical relationship is implemented in a simulation model and added to the solution element of the *Centrifugal pump*. The calculated critical rotational speed and the actual simulated speed can then be linked to the design requirement and compared there.

The two procedures described above ensure that requirements can be checked both at the system level and in a solution element in such a way that no additional modeling effort is required for each solution combination generated. Sect. 4.1 thus fully demonstrated how alternative technical solutions can be modeled in such a function-oriented manner that solution combinations can be efficiently formed, simulated and evaluated in the three following subchapters.

4.2 Composition of solution combinations









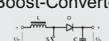











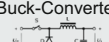



On the basis of the functionally structured modeled solutions (cf. chapter 4.1), a solution combination can be generated. For this purpose, a new system solution is created

which specifies the abstract system solution via a generalization relation and inherits all its properties. Subsequently, the individual inherited abstract solutions can be redefined. For each inherited abstract solution, one of the specific concrete solutions can be selected. For example, the *Centrifugal pump* can be selected as specification for the abstract solution element *Apply mechanical energy to fluid*. If a concrete solution is selected for each of the abstract solutions, a solution combination is composed (Fig. 6a). Some tools for the application of MBSE like CATIA Magic supports the selection of a solution by actively suggesting all modeled specifications. Often only one click per solution is needed to generate a solution combination. This minimizes the modeling effort and no new elements or relationships have to be modeled or adjusted (Fig. 6b). Since only specified variants of the respective abstract solution and therefore of the function can be selected, the functional compatibility of all possible solution combinations according to the function architecture, is always ensured.

In preparation of the following simulation, the solution combination has to be parameterized. For this purpose, one of the instantiated parameter sets can be selected for each active surface set. Again, the tool provides active support by proposing all parameter sets of an active surface set and allows them to be integrated into the model of the solution combination with a single mouse click.

With the described methods, solution combinations can be created, which can then be efficiently simulated and evaluated. The number of possible overall solutions can quickly explode combinatorically (cf. Sect. 1). Compared to the state of research the method developed in this paper therefore supports the generation of a high number of solution combinations by efficient modeling. The method does not force the generation of every possible solution combination, but this is theoretically possible for small functional architectures. For example, it is possible to selectively limit the number of solutions generated by experts and to verify and evaluate only promising solution combinations in

Fig. 7 Verification and evaluation of solution combinations for requirement set a (a) and an alternative requirement set b (b)

a Verification and evaluation of solution combinations for requirement set a and ...								b set b
#	Scale electric Energy 	Convert electric to mech. energy 	Conduct mech. energy 	Apply mech. energy to fluid 	Cavitation-free $n < n_{krit}$	Max. Power $P \leq 2000 \text{ W}$	Efficiency η_a	Efficiency η_b
1	Boost-Converter 	DC Shunt Motor 	Cylinder interface fit 	Impeller Pump 	✓	✓	15%	10%
2	Boost-Converter 	DC Shunt Motor 	Key Connection 	Centrifugal Pump 	✓	✓	75%	5%
3	Boost-Converter 	DC Shunt Motor 	Splined shaft connection 	External Gear Pump 	✓	✗	10%	27%
4	Boost-Converter 	DC Shunt Motor 	Cylinder interface fit 	Axial Piston Pump 	✓	✗	2%	30%
72	Buck-Converter 	DC Shunt Motor 	Cylinder interface fit 	Centrifugal Pump 	✗	✗		

Requirement set a
 $\dot{V} = 360 \text{ L/min}$ $\Delta p = 0.4 \text{ bar}$ $U = 12.5 \text{ V}$ $\rho = 997 \text{ kg/m}^3$ $\eta = 1 \text{ mPa}\cdot\text{s}$

Requirement set b
 $\dot{V} = 24 \text{ L/min}$ $\Delta p = 4 \text{ bar}$ $U = 12.5 \text{ V}$ $\rho = 900 \text{ kg/m}^3$ $\eta = 80 \text{ mPa}\cdot\text{s}$

a model-based manner. With this approach, of course, there is a risk of overlooking the best possible solution combination. If a brute force approach is chosen, automation is certainly required to generate, simulate and evaluate all solution combinations. While in the example presented only some exemplary requirements were checked, even in early phases of an industrial development process more requirements have to be validated and more models have to be used. Therefore, in the brute force approach, computing capacity will be the limiting resource [26].

A promising approach between brute force and heuristic single consideration could be algorithms from mathematics for the control of the combinatorial explosion. The transfer of these algorithms to the combination of solutions in function-oriented product development should therefore be further explored in the future.

Using the example of the electric coolant pump, the “brute-force-method” was used to form every possible solution combination. According to this procedure, 72 possible solution combinations can be formed. The subsequent combination of the parameter sets was done heuristically. According to the “brute-force-method”, 625 parameter set combinations are conceivable for each solution combination. This results in a total of 45,000 theoretically possible variants that cannot be handled. Using a heuristic and iterative procedure for the parameterization of each solution combination in combination with the simulation and evaluation, exactly one best possible parameterization is determined for each solution combination. In total, 72 parameterized solution combinations were formed (Fig. 7a).

4.3 Simulation

To simulate a parameterized solution combination, adequate transformations of the 1d simulation models for each solution must be selected. When selecting each transformation, it is important to ensure that an executable workflow is created in the context of the given boundary conditions, so that each 1d simulation model can be executed in a correct way. The physical behavior of the overall system can then be simulated without additional modeling effort, simply by selecting suitable model transformations.

For the example of the electric coolant pump, the selection of suitable transformations was done manually. The sequence within the workflow and the respective transformations of the 1d simulation models were the same for each solution combination. Solvers for the calculation and transformation are available in explicit simulation programs, such as the program Simscape from MathWorks [25]. These programs are able to select valid transformations of the mathematical relationships by themselves. For the successful application of the elaborated method in industry and for more complex products, it must be flanked by suitable solvers. Alternatively, a decoupling of the simulation into more appropriate programs is also conceivable. Both approaches need to be further explored in the future. In the example of the electric coolant pump, the 1d simulation models considered only static behavior. The representation of dynamic physical behavior could be an important extension in the future.

4.4 Verification und evaluation

In this chapter we will explain how solution combinations can be initially verified and evaluated against design requirements based on the simulation results (cf. Sect. 4.3). Due to the previously set and inherited requirement links, the verification of parameterized solution combinations can be executed parallel to the simulation and without any further action in user tools like CATIA Magic [26]. The electrical current required to reach the operating point and the resulting electrical power were calculated. If the power exceeds 2000 watts, the corresponding design requirement is not fulfilled. In total, 16 of the 72 possible solution combinations meet both requirements concerning the maximum electrical energy as well as a cavitation-free behavior. The other 56 solution combinations require a higher power or may cause cavitation (Fig. 7a).

For the selection of the best possible out of 16 verified solution combinations, an additional evaluation is needed to compare the solution combinations with each other. Here, the efficiency is used as an exemplary evaluation criterion. The efficiency can be calculated from the electrical power input and the incoming and outgoing volume flows of the coolant. In this application, the best possible solution combination with an efficiency of 75% is a solution combination composed of a *Boost converter*, *DC shunt motor* and *Centrifugal pump*. The selection of the solution element with which mechanical energy is conducted has no influence on the overall efficiency, since the efficiency of these solution elements was always assumed to be 100%. In addition to the values listed here, all flow variables of the entire system were simulated. The analysis of this data suggests that an even better efficiency could be achieved by adding a mechanical gearbox between the motor and the pump.

Furthermore, we can demonstrate that the models used react sensitively to requirement changes. With the reduction of the required volume flow, the increase of the differential pressure and a different fluid, instead of cooling water here an engine oil, the calculated efficiencies and the fulfillment of the requirements turn out differently. The positive displacement pumps perform much better (Fig. 7b).

5 Conclusion

In this paper, a method was presented which enables the systematic and function-oriented composition, simulation and evaluation of solution combinations. All research hypotheses (cf. Sect. 3) could be confirmed. The approach according to Jacobs et al. (known as motego method) [2] was applied and extended with bidirectional solution flows, abstract solutions and 1d simulation models regarding the physical effect and efficiency for the simulation of solution

combinations. The method was tested for the exemplary development of an electromechanical coolant pump (Fig. 7a). The paper points out research questions which have to be solved in order to improve the method for future testing and a future industrial application.

There is the potential to extend the method with appropriate simulation models, so that additional requirements, such as for a dynamic system behavior, functional safety (e.g. for conduct mechanical energy), costs, design space, etc., can be verified and evaluated in a Multidisciplinary Design Analysis and Optimization (MDAO).

Even though it could be shown that the method can already consider a comparatively large number of solution combinations, the combinatorial explosion of large functional architectures still represents a major challenge. The method significantly reduces the modeling effort required to generate many possible solutions. The remaining modeling effort is mainly necessary for the generation of the individual functions, solutions and simulation models. The use of model based function and solution libraries [2, 5] promises an enormous reduction of these remaining efforts and promises a very efficient solution combination directly out of a library. With the integration of efficient optimization methods into the approach shown here, the composition and evaluation of a very large number of solution combinations could be possible. Thus, the paper makes an essential contribution to the model-based and function-oriented consideration of a broad field of possible solutions targeting to find the best possible combination.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest L. Irnich, G. Jacobs, T. Zerwas and C. Konrad declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bender B, Gericke K (2021) Pahl/Beitz Konstruktionslehre, 9th edn. Springer, Berlin
2. Jacobs G, Konrad C, Berroth J, Zerwas T, Höpfner G, Spütz K (2022) Function-oriented model-based product development. In: Krause D, Heyden E (eds) Design methodology for future products: data driven, agile and flexible, 1st edn. Springer, Cham

3. Koller R (2013) Konstruktionslehre für den Maschinenbau. Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen, 4th edn. Springer, Berlin, Heidelberg
4. Koller R, Kastrup N (1998) Prinziplösungen zur Konstruktion technischer Produkte, 2nd edn. Springer, Berlin
5. Zerwas T, Jacobs G, Spütz K, Höpfner G, Drave I, Berroth J, Guist C, Konrad C, Rumpe B, Kohl J (2021) Mechanical concept development using principle solution models. IOP Conf Ser: Mater Sci Eng 1097(1):12001
6. Incose (2007) Systems Engineering Vision 2020. https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf. Accessed 4 Oct 2022
7. Höpfner G, Jacobs G, Zerwas T, Drave I, Berroth J, Guist C, Rumpe B, Kohl J (2021) Model-based design workflows for cyber-physical systems applied to an electric-mechanical coolant pump. IOP Conf Ser: Mater Sci Eng 1097(1):12004
8. Weikiens T (2020) SYSMOD—the Systems modeling toolbox. Pragmatic MBSE with SysML, 3rd edn. MBSE4U booklet series; MBSE4U. Fredesdorf
9. Pohl K, Hönninger H, Achatz R, Broy M (2020) Model-based engineering of embedded systems. The SPES 2020 Methodology. Springer, Berlin
10. Eigner M, Dickopf T, Schulte T, Schneider M (2016) mecPro 2 – Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysM. In: Schulze S-O, Muggeo C, Abulawi J (eds) Tag des Systems Engineering Ulm, 11.–13. November 2015 Hanser, München, pp 163–172
11. Kruse B (2017) A library-based concept design approach for multi-disciplinary systems in SysML. ETH, Zurich
12. Kruse B, Shea K (2016) Design library solution patterns in SysML for concept design and simulation. Procedia CIRP 50:695–700
13. Moeser G, Kramer C, Grundel M, Neubert M, Kümpel S, Scheithauer A, Kleiner S, Albers A (2016) Fortschrittsbericht zur modellbasierten Unterstützung der Konstrukteurstätigkeit durch FAS4M. In: Schulze S-O, Muggeo C, Abulawi J (eds) Tag des Systems Engineering Ulm, 11.–13. November 2015 Hanser, München, pp 69–78
14. Zingel C, Albers A, Matthiesen S, Maletz M (2012) Experiences and advancements from one year of explorative application of an integrated model-based development technique using C&C²-A in SysML. IAEG Int J Comput Sci 39(2):165–181
15. Spütz K, Berges J, Jacobs G, Berroth J, Konrad C (2022) Classification of simulation models for the model-based design of plastic-metal hybrid joints. Procedia CIRP 109:37–42
16. Spütz K, Jacobs G, Zerwas T, Konrad C (2023) Modeling Language for the Function-oriented Development of Mechatronic Systems with motego. Eng Res Ingenieurwes. <https://doi.org/10.1007/s10010-023-00623-4>. (accepted, but not yet published)
17. Habermehl C, Höpfner G, Berroth J, Neumann S, Jacobs G (2022) Optimization workflows for linking model-based systems engineering (MBSE) and multidisciplinary analysis and optimization (MDAO). Appl Sci 12(11):5316
18. Object Management Group (2017) OMG system modeling language specification: version 1.5. <https://www.omg.org/spec/SysML/1.5>. Accessed 14 July 2022
19. Zerwas T, Jacobs G, Kowalski J, Husung S, Gerhard D, Rumpe B, Zeman K, Vafaei S, König F, Höpfner G (2022) Model signatures for the integration of simulation models into system models. Systems 10(6):199
20. Schulz H (1977) Die Pumpen. Arbeitsweise, Berechnung, Konstruktion, 13th edn. Springer, Berlin, Heidelberg, New York
21. Simulink (2022) Simulation und Model-Based Design. <https://de.mathworks.com/products/simulink.html>. Accessed 25 Nov 2022
22. (2022) Cameo systems modeler—CATIA—Dassault systèmes. <https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/>. Accessed 25 Nov 2022
23. Gülich JF (2010) Kreiselpumpen. Handbuch für Entwicklung, Anlagenplanung und Betrieb. Springer, Berlin
24. Wesche W (2016) Radiale Kreiselpumpen. Berechnung und Konstruktion der Hydrodynamischen Komponenten, 2nd edn. VDI-Buch. Springer, Berlin, Heidelberg
25. (2022) Simscape. <https://de.mathworks.com/products/simscape.html>. Accessed 25 Nov 2022
26. No Magic (2022) Cameo systems modeler documentation. <https://docs.nomagic.com/display/CSM190SP4/Cameo+Systems+Modeler+Documentation>. Accessed 14 July 2022
27. Hoffman KL, Padberg M (2001) Traveling salesman problem. In: Gass SI, Harris CM (eds) Encyclopedia of Operations Research and Management Science. Springer, New York, NY. https://doi.org/10.1007/1-4020-0611-X_1068
28. Tecomotive (2022) Pierburg “CWA400”. <https://www.tecomotive.com/de/produkte/CWA400.html>. Accessed 8 Jan 2023