Forsch Ingenieurwes (2023) 87:387–398
https://doi.org/10.1007/s10010-023-00623-4

ORIGINALARBEITEN/ORIGINALS

Modeling language for the function-oriented development of mechatronic systems with motego

Kathrin Spütz[1] · **Georg Jacobs**[1] · **Thilo Zerwas**[1] · **Christian Konrad**[1]

Received: 31 October 2022 / Accepted: 19 January 2023 / Published online: 20 March 2023

Abstract
As mechatronic products gain in popularity, methods for mastering the complexity of these systems in development become increasingly relevant, such as model-based systems engineering (MBSE). Main pillars of MBSE are method, language and tool. A method specifies procedures in product development. The application of the method is supported by a language and tool as the language specifies a system of symbols with which development artifacts can be represented in a software environment (i.e. tool). Currently, various MBSE methods exist, such as motego. Motego specifies a framework for the function-oriented seamless development of mechatronic systems from requirements to the physical realization down to mechanical and electrical contacts and the description of these via parameters and models. Central element in MBSE is the system model, which connects all relevant development artefacts. The system model is created with a language in a software environment such as Cameo Systems Modeler. In MBSE, the graphical systems modeling language SysML is widely established. The language elements in SysML are very abstract and numerous. As a result, the language is difficult to apply. However, its reasonable applicability is an essential prerequisite for the introduction of the motego methods in industrial practice. This results in the following research need: A specific modeling language for the motego method is needed that supports its reasonable application. Therefore, in this paper a modeling language is presented whose language elements are specifically adapted to the motego method. With the help of this domain specific language, the user is guided through method-compliant modeling.

✉ Kathrin Spütz
kathrin.spuetz@imse.rwth-aachen.de

[1] Institute for Machine Elements and Systems Engineering, RWTH Aachen University, Schinkelstr. 10, 52062 Aachen, Germany

Modellierungssprache für die funktionsorientierte Entwicklung von mechatronischen Systemen mit motego

Zusammenfassung
Mit der zunehmenden Verbreitung mechatronischer Produkte gewinnen Methoden, wie die modellbasierte Systementwicklung (MBSE), zur Komplexitätsbeherrschung in der Produktentwicklung an Bedeutung. Die wichtigsten Säulen der MBSE sind die Methode, die Sprache und das Werkzeug. Die Methode spezifiziert das Vorgehen in der Produktentwicklung. Die Anwendung der Methode wird durch die Sprache und das Werkzeug unterstützt, indem die Sprache ein System von Symbolen spezifiziert, mit dem die Entwicklungsartefakte in der Softwareumgebung (d.h. dem Werkzeug) dargestellt werden. Zurzeit gibt es verschiedene MBSE-Methoden, wie beispielsweise motego. Die motego Methode spezifiziert ein Rahmenwerk für die funktionsorientierte, durchgängige Entwicklung mechatronischer Systeme von Anforderungen bis zur physikalischen Realisierung auf Systemebene und auf Ebene der mechanischen und elektrischen Kontakte. Die Beschreibung und Verknüpfung aller relevanten Entwicklungsartefakte erfolgt dabei in einem zentralen Systemmodell über Parameter und ausführbare Modelle. Dieses Systemmodell wird in der Regel mit der grafische Systemmodellierungssprache SysML in einer Softwareumgebung wie dem Cameo Systems Modeler erstellt. Die Sprachelemente der SysML sind sehr abstrakt und zahlreich. Daher ist die Sprache schwer anwendbar. Ihre sinnvolle Anwendbarkeit ist jedoch eine wesentliche Voraussetzung für die Einführung der motego Methoden in der industriellen Praxis. Daraus ergibt sich der folgende Forschungsbedarf: Es wird eine spezifische Modellierungssprache für die motego Methode benötigt, die deren sinnvolle Anwendung unterstützt. Hierzu wird in dieser Arbeit eine speziell auf die motego Methode abgestimmte Modellierungssprache vorgestellt, die den Anwender durch die methodenkonforme Modellierung leitet.

1 Introduction

As mechatronic products gain in popularity [1], methods for mastering the complexity of these systems in development become increasingly relevant, such as model-based systems engineering (MBSE) [2]. Pillars of MBSE are the method used to develop or analyze the technical system as well as the language and the tool used to digitally represent the system. Several MBSE methods exist. Among these methods, the motego method is particularly charac-

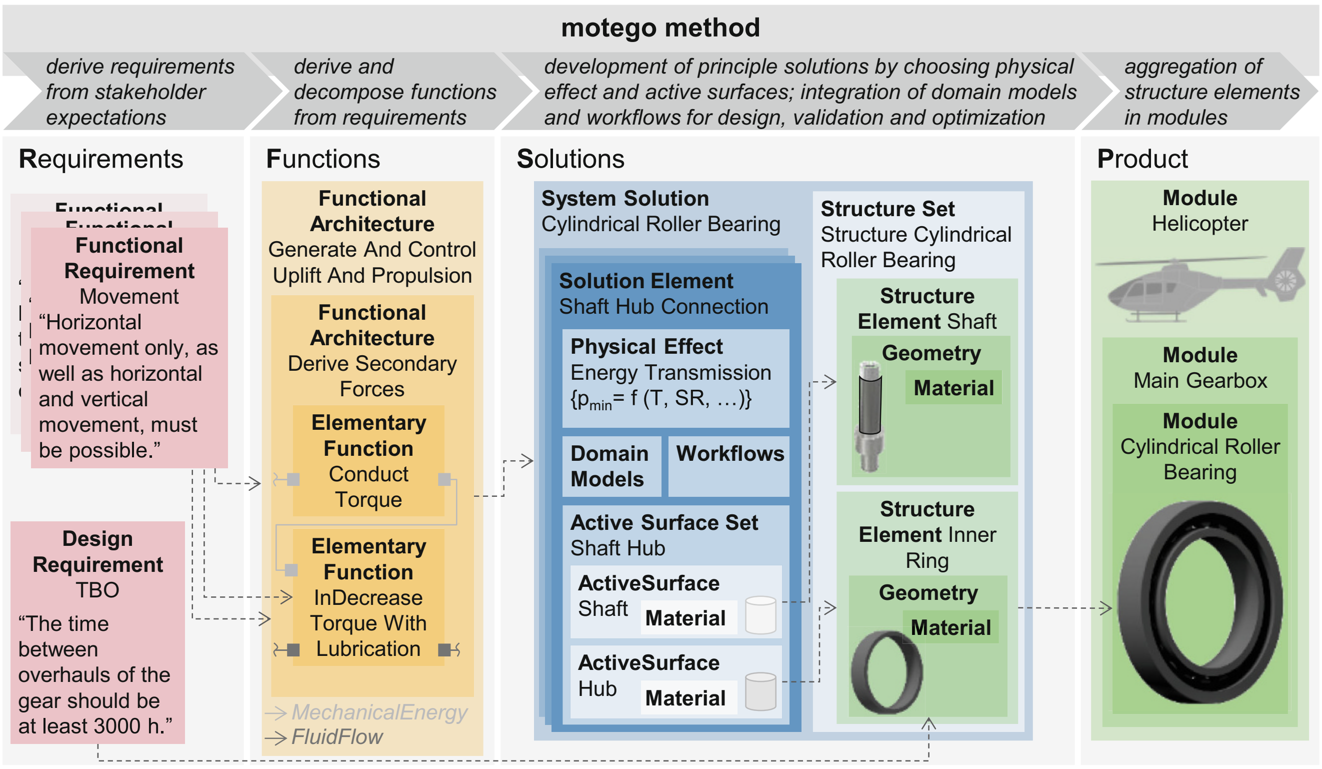
motego method
derive requirements from stakeholder expectations
derive and decompose functions from requirements
development of principle solutions by choosing physical effect and active surfaces; integration of domain models and workflows for design, validation and optimization
aggregation of structure elements in modules
Requirements
Functional Requirement
Movement
"Horizontal movement only, as well as horizontal and vertical movement, must be possible."
Design Requirement
TBO
"The time between overhauls of the gear should be at least 3000 h."
Functions
Functional Architecture
Generate And Control Uplift And Propulsion
Functional Architecture
Derive Secondary Forces
Elementary Function
Conduct Torque
Elementary Function
InDecrease Torque With Lubrication
MechanicalEnergy
FluidFlow
Solutions
System Solution
Cylindrical Roller Bearing
Solution Element
Shaft Hub Connection
Physical Effect
Energy Transmission
{p_min = f (T, SR, …)}
Domain Models
Workflows
Active Surface Set
Shaft Hub
ActiveSurface
Shaft
Material
ActiveSurface
Hub
Material
Structure Set
Structure Cylindrical Roller Bearing
Structure Element Shaft
Geometry
Material
Structure Element Inner Ring
Geometry
Material
Product
Module
Helicopter
Module
Main Gearbox
Module
Cylindrical Roller Bearing

Fig. 1 Function-oriented model based development with the motego method using the example of a helicopter following [3, 5, 7, 10]

terized by its function orientation, formalized description of solutions down to the contact level, along with seamless integration and execution of simulation models. Especially in the mechanical domains, considering the contact level is an essential advantage of the motego method, since the functional implementation occurs primarily in the contact of component surfaces to each other.

Motego method is suitable for different objectives: For the development of complex mechatronic systems based on the stakeholder's expectations and for the analysis of complex problems. An example application is the analysis of wear in a helicopter gearbox during operation. Although various simulation models for predicting wear in different mechanical contacts of the gearbox already exist, the combination into a systemic wear model, including the consideration of interactions, has not yet been examined. The motego method with its formal description of the technical system and its relationships over several hierarchical levels is suitable for modeling these relationships. Especially the breakdown to contact level and the seamless connection of simulation models with mechanical and electrical contacts is valuable for wear analysis in technical systems, because the wear does not occur in components but in the contact of two component surfaces to each other.

Figure 1 illustrates the main pillars of mechatronic product development and analysis using the motego method, exemplified by a helicopter drive train gearbox. Definition of requirements is the basis for system development and analysis. Requirements capture the expectations of all stakeholders regarding the system. Starting from the functional requirements, functions are derived, decomposed and combined with further functions to create a consistent function model. Core of function-oriented model-based product development with the motego method is this function model, since it describes the intended systems behavior solution-neutral. This solution-neutral description serves as a communication means between domains, as it defines the interfaces between engineering, software and electronic domains. By function-oriented encapsulation as well as specifying explicit interfaces between the domains in the motego method, the parallel agile development of mechatronic products is facilitated.

In the case of the helicopter gearbox, the functional model is used for the systematic breakdown of the overall system down to contact level and the specification of the energy, material and signal flows between them. Thus, the functional model provides the structure to systematically connect simulation models of individual subsystems and to specify parametric relationships in the energy, material and signal flows between them.

Next step of the motego method is the domain specific, conceptual description of the function's realization. The challenge in the conceptual description of mechanical solutions lies primarily therein that a function is not realized directly via a separable part of the product, but rather the parts indirectly fulfill the overall function as a whole via existing physical effects. Components, as part of the final product and the smallest unit that can be manufactured in mechanics, are typically involved in the fulfillment of several functions. An important challenge in mechanics is the modeling of these n:m relationships between functions and components. In contrast to the informatics domain, where a function can be realized directly by a separate part of code, i.e. product. In order to consistently map the relationships between functions and components in the mechanic domain, the motego method describes the solutions down to the contact level with so-called solution elements. The solution element describe conceptually the realization of functions with a physical effect and the active surfaces geometry and material where this effect occurs [3]. The physical effect is described with executable simulation models in expert tools such as MATLAB [4], which are linked in the system model to a parametric description of the functional flows and active surfaces. The modeling of physical effects and active surfaces is described in detail in [5] using the example of volume displacement generated by a centrifugal pump wheel. For the design, validation and optimization of the solutions, the solutions are expanded by domain models and workflows which control these [3, 6–9].

Part of the gearbox solution model is the "Shaft Hub Connection" between the shaft and the inner bearing ring. This contact realizes the elementary function "Conduct Torque". The mechanical energy transmitted via the shaft hub connection is directly related to the contact between the inner bearing ring tread and the balls of the roller bearing. Since this contact realizes the function "InDecrease Torque With Lubrication" this contact and the shaft-hub contact are part of the higher-level system solution "Cylindrical Roller Bearing". By parametrically characterizing the geometry and materials of the two contacts and modeling the physical effect as well as domain models, the relationship between the energy transmission in the shaft-hub connection and the generated wear in the bearing can be sufficient described. Similar to the two contacts described, all wear-relevant contacts of the gearbox need to be described and related to each other in order to determine the wear of the entire system.

At higher system levels, the solution elements are aggregated to form system solutions. In addition, at system level, the active surfaces from the solution elements are aggregated to structure elements. This structure elements link the function-oriented developed solutions to the final product design in the product model. Within the product layer, the entire product is aggregated into modules up to the product [10]. Main benefit of this layer is to structure the product's parts in different views such as the production and controlling view. These different structures are established simultaneously and serve as a reference for further development phases, such as the product's production and the purchase of raw materials. By providing these preliminary structures, information can be passed on to downstream process steps at an early stage in the development process, and knowledge gained in these steps can be returned as new requirements to the development process. By providing feedback from the downstream process steps to development, restrictions can be considered at an early stage, when the costs of change are still low. For the wear analysis in the helicopter gearbox, product layer is not significantly relevant, since the wear in the contacts is caused by mechanical friction of the component surfaces on each other and not in the structures. Furthermore, the higher-level systems have no direct influence on the wear, but only indirect influence via the definition of functional flows and bearing requirements.

For the development and analysis of the technical systems, the motego method described here provides a procedure and architecture. However, in order to map the architecture digitally, a language and a tool, i.e. a software environment in which the language is implemented, are also required. The suitability of this modeling language is a critical enabler for MBSE [11]. Since the mapped technical systems are intrinsically complex, the meaningful applicability of the modeling language is largely determined by the ability to both efficiently and consistently map the development artifacts specified in the method at a sufficient level of granularity.

2 State of research

The graphical general-purpose system modeling language (SysML) [12] is widely used for modeling architectures in the context of MBSE [2]. SysML was developed based on the unified modeling language (UML) [13] and extends it with language elements such as requirements, especially designed for modeling in the context of MBSE [12, 14]. SysML is characterized by its generic character and can therefore be used method-independently in MBSE [11]. SysML includes language elements (notation) and relationships (syntax), as well as diagrams that can be used to represent technical systems. The challenge in modeling technical systems with SysML is that the SysML notation and syntax are very abstract and numerous. Consequently, the language is difficult to understand and not efficiently applicable to model architectures with a specific MBSE method.

To enhance the challenging applicability of the SysML, for the model-based product development methods FAS4M [15, 16], SYSMOD [17], and SPES [18] the domain-specific modeling languages MechML [15, 16, 19], SYSMOD [20], and SPESML [21] exist. The notation and syntax of these domain-specific modeling languages is defined on the basis of the established SysML resp. UML notation and syntax [2]. For the definition of domain-specific notation and syntax, UML provides dedicated profile mechanisms. With the help of these mechanisms, refinements of UML, such as SysML, can be implemented as profiles in MBSE tools. Domain-specific modeling languages MechML [15, 16, 19], SYSMOD [20] and SPESML [21] use these mechanisms to further specify SysML resp. UML for the devel-

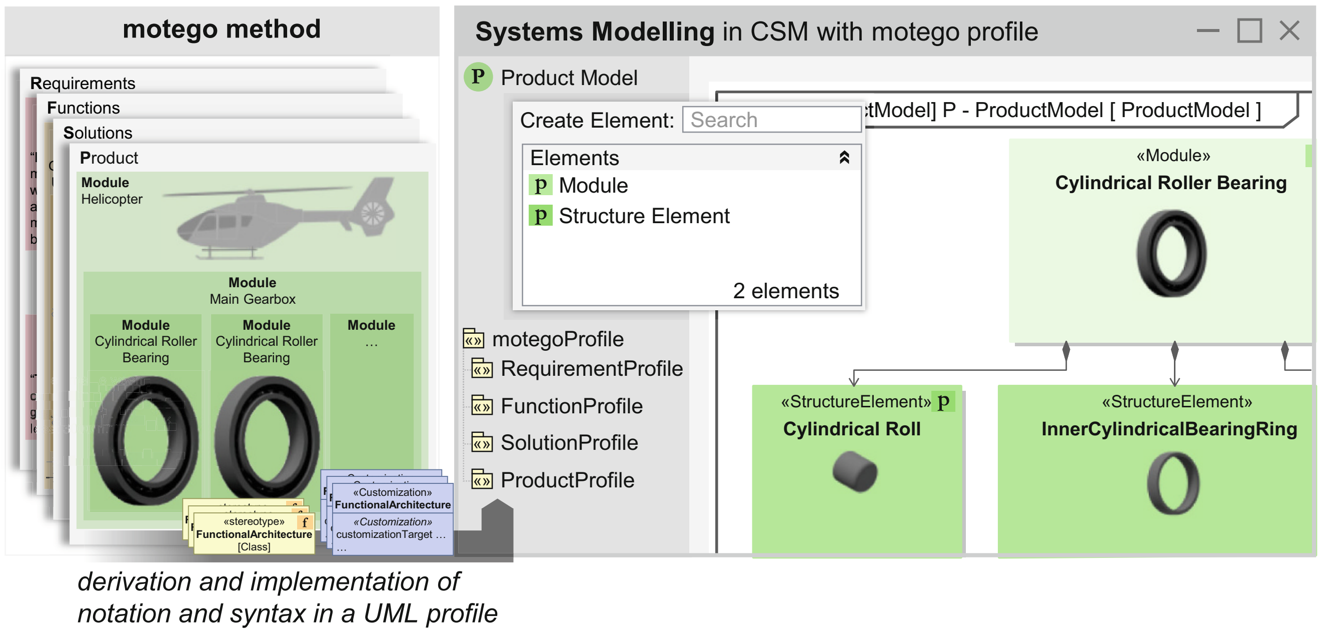

Fig. 2 Development and usage of domain specific language notation and syntax for the motego method in Cameo Systems Modeler (CSM)

opment of mechatronic systems and thus make the notation and syntax more comprehensible for the user.

MechML is a SysML language profile specially developed for the FAS4M method, which extends SysML with language elements that are necessary for the FAS4M method. Since the method essentially differs from the motego method in the logic layer, there are no suitable language elements available in MechML for the solution layer of the motego method. On the one hand, there are no language elements for the parametric, executable modeling of the principle solutions as a subset of the solution elements. On the other hand, there are no language elements for the establishment of model networks and their semi-automated execution as shown in [7, 9, 22].

The SYSMOD profile [20] is designed for the SYSMOD method and contains special language elements for this method. The focus of the profile lies on model elements for requirements management and modeling system architectures. For this reason, it does not contain any language elements for the modeling of executable simulation models and controlling of these. As a result, the SYSMOD profile cannot sufficiently address the solution layer of the motego method either.

SpesML [18] and the associated SPES method mainly support the development of software intensive systems such as cyber-physical systems as well as mere software systems. The modeling of mechanical solution concepts and the description of their behavior with simulation models is not considered. However, this is essential for the motego method to develop its full potential.

3 White spot and research questions

Summarizing, the modeling languages presented in the state of the art improve the user friendliness for the respective MBSE methods. Since the methods differ significantly from the motego method in the logic layer, they do not provide sufficient support for the development executable mechanical solutions down to contact level. Thus, the applicability of these domain-specific modeling languages for the motego method is not given. As a result, there is no general language standard such as SysML as well as no domain-specific modeling languages in the state of the art that enable reasonable application of the motego method.

The state-of-the-art results in the need for a domain-specific modeling language (DSL) specially developed for the motego method, which enables the reasonable, standardized application of the method. In order to achieve this objective, the following research question must be answered: Which language elements (notation) and relationships (syntax) are required for the efficient and consistent implementation of system architectures with the motego method?

4 Method

In order to develop a domain-specific language for the motego method, it is necessary to identify method elements which need to be represented in the DSL. Based on the identified elements, the notation and syntax are derived and implemented in the software environment Cameo Systems Modeler (CSM) [23] (cf. Figure 2). For implementing specific notation and syntax for the motego method, UML offers favorable prerequisites with the existing profile mechanisms. With the help of these profile mechanisms, SysML itself is already implemented in modeling environments, so that the language elements and relationships of the motego modeling language can also be implemented as a refinement of SysML in a motego profile. This is particularly valuable because mechanisms that have already been implemented can be used, and the resulting system architectures are still compatible with SysML.

A major obstacle to the efficient and consistent use of SysML is the large number of modeling options available to the user. To overcome this problem, the implementation of the DSL for the motego method includes, dynamic, context-specific adaptation of the modeling possibilities. This means that, depending on the method layer and the language elements, only those elements and relationships are suggested to the user which are proposed in the motego method. In Fig. 2 dynamic, context-specific adaptation of the modeling possibilities are illustrated by the example of the product layer where instead of more than fifty SysML elements only the two elements *module* and *structure element* are suggested to the user, which are part of the product layer of the motego method. This guides the user through each step of the motego method and thus significantly sup-

Table 1 Elements of motego requirement layer needed for helicopter gearbox wear analysis

Functional Requirement	*Objective: System Load Cases*
Functional requirements precisely specify with state machines desired system states as part of the architecture [3, 5]	To specify environmental influences for the gearbox during operation and thus for simulation of wear, the entire system with its load cases need to be considered
Design Requirement	*Objective: System Restrictions*
Design requirements describes verifiable by using a predefined glossary [24] restrictions for system parameters [3, 5, 8]	In order to validate the helicopter gearbox design, it is necessary to specify technical and economic system restrictions like maximum wear or minimum lifetime
Requirement Model	*Objective: Directory*
A requirement model serves as a kind of folder where all elements of the requirements layer, i.e. requirements, related tables, and diagrams are stored	To collect all requirements for the helicopter gearbox as well as the tables and diagrams in which they are presented a kind of directory is needed

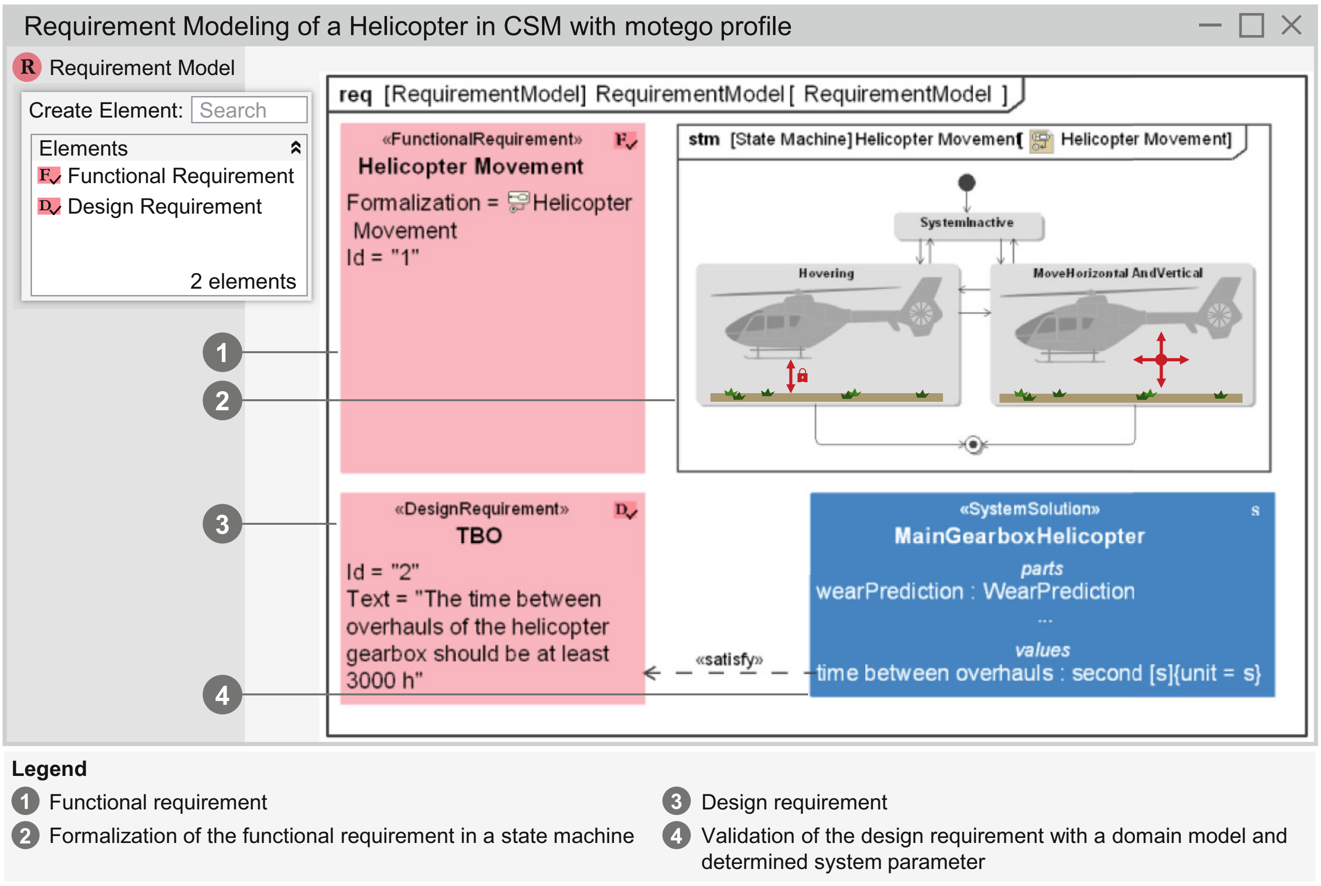
Requirement Modeling of a Helicopter in CSM with motego profile
Requirement Model
Create Element: Search
Elements
Functional Requirement
Design Requirement
2 elements
req [RequirementModel] RequirementModel [RequirementModel]
«FunctionalRequirement»
Helicopter Movement
Formalization = Helicopter Movement
Id = "1"
stm [State Machine] Helicopter Movement[Helicopter Movement]
SystemInactive
Hovering
MoveHorizontal AndVertical
«DesignRequirement»
TBO
Id = "2"
Text = "The time between overhauls of the helicopter gearbox should be at least 3000 h"
«satisfy»
«SystemSolution»
MainGearboxHelicopter
parts
wearPrediction : WearPrediction
...
values
time between overhauls : second [s]{unit = s}
1
2
3
4
Legend
1 Functional requirement
2 Formalization of the functional requirement in a state machine
3 Design requirement
4 Validation of the design requirement with a domain model and determined system parameter

Fig. 3 Requirement modeling of a helicopter in CSM with motego profile

ports the user in the efficient and consistent application of the motego method.

4.1 Requirements

For analyzing and developing technical systems, requirements must first be specified. Requirements in terms of this paper define expectations for the project from the point of view of all stakeholders. Capturing the requirements encourages a common understanding of the objectives against which the project results can be continuously reviewed. The specific objectives for which requirements are used in the context of developing an architecture for systemic wear analysis in a helicopter gearbox are outlined in Table 1. Furthermore, elements of the motego method that can be

Table 2 Elements of motego functions layer needed for helicopter gearbox wear analysis

Elementary Function	*Objective: Contact Description*
Contacts are described in a solution-neutral way via so-called elementary functions. The elementary functions specify the incoming and outgoing energy, material, and signal flows that are conveyed and transformed in the contact [3, 5]	Wear occurs in the contacts. To build up the structure for connecting the wear models of the different contacts, these must be represented in a solution-neutral way
Functional Architecture	*Objective: Structure Contacts*
Aggregation of multiple functions and characterization of their interactions is specified with functional architectures [3, 5]	To structure contacts for better understanding, functions composed of further functions have to be mapped
Function Flows	*Objective: Map Function Interactions*
Interactions between functions are specified with energy, material and signal function flows [3, 5, 28]. Function flows are quantified by concrete parameters values [3, 5]	To map the relationship between the contacts and subsystems the interactions between functions have to be mapped
Function Model	*Objective: Directory*
A function model serves as a kind of folder where all elements of the functions layer, i.e. elementary functions and functional architectures, function flows, related tables, and diagrams are stored	To collect all functions and function flows for the helicopter gearbox as well as the tables and diagrams in which they are presented a kind of directory is needed

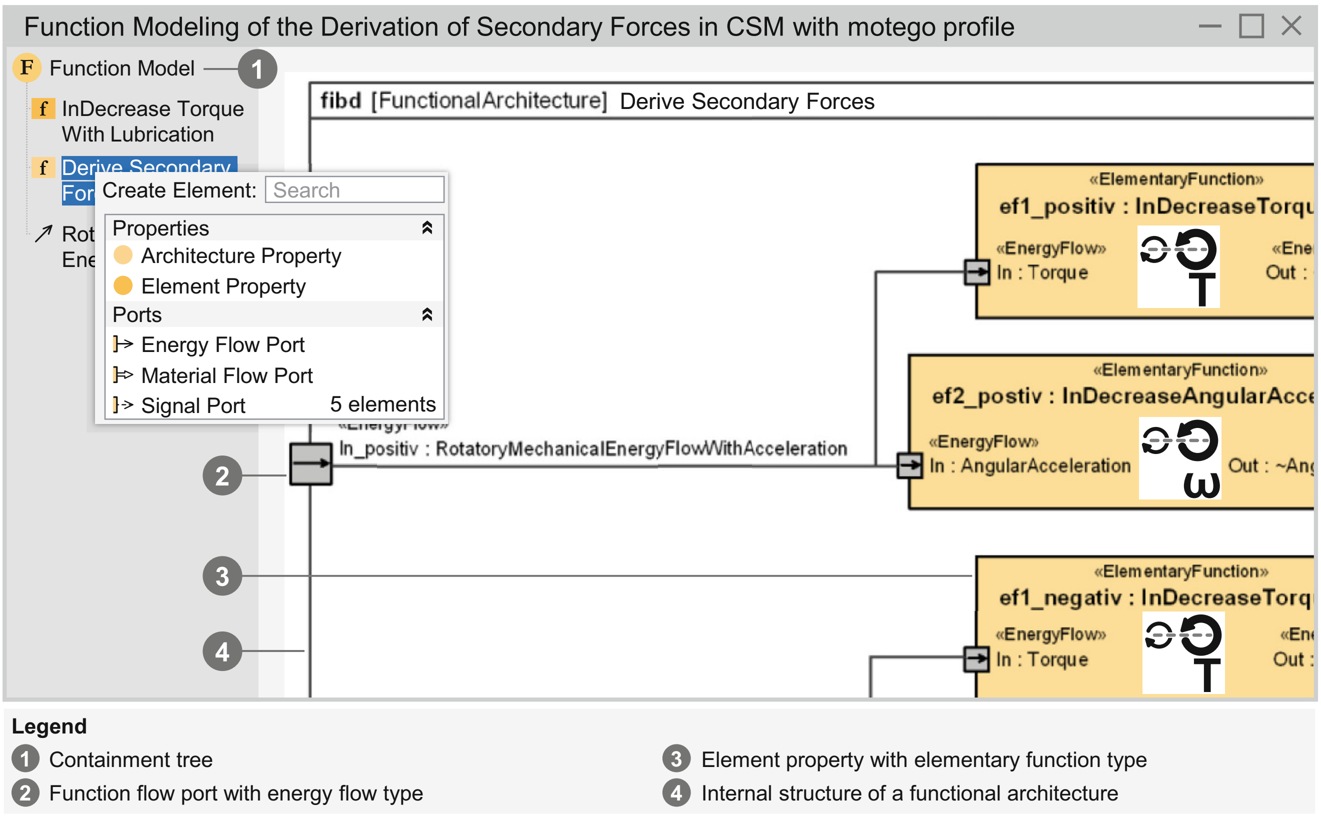
Function Modeling of the Derivation of Secondary Forces in CSM with motego profile
F Function Model
1
f InDecrease Torque With Lubrication
f Derive Secondary For
Rot Ene
Create Element: Search
Properties
Architecture Property
Element Property
Ports
Energy Flow Port
Material Flow Port
Signal Port 5 elements
fibd [FunctionalArchitecture] Derive Secondary Forces
«EnergyFlow»
In_positiv : RotatoryMechanicalEnergyFlowWithAcceleration
«ElementaryFunction»
ef1_positiv : InDecreaseTorqu
«EnergyFlow»
In : Torque
Out :
«ElementaryFunction»
ef2_postiv : InDecreaseAngularAcce
«EnergyFlow»
In : AngularAcceleration
Out : ~Ang
«ElementaryFunction»
ef1_negativ : InDecreaseTorq
«EnergyFlow»
In : Torque
Out :
2
3
4
Legend
1 Containment tree
2 Function flow port with energy flow type
3 Element property with elementary function type
4 Internal structure of a functional architecture

Fig. 4 Excerpt of the function model of the derivation of secondary forces in CSM with motego profile

used to map these objectives within the architecture are described.

Summarizing, the following elements need to be implemented in the motego profile—*functional requirement, design requirement* and *requirement model*. A functional requirement specifies the intended functionality of a technical system [5]. Through the formalization of this requirement as a state machine preconditions for automated generation of test cases are established [25, 26]. Design requirements describe boundary conditions of parameters occurring in the system [5].

Figure 3 shows how the implemented motego profile with these elements is used for modeling requirements in CSM. The implemented motego profile allows the requirement layer to be implemented as part of the architecture in the system model (requirement model in Fig. 3). As subordinate elements to the requirement model, only the two requirement types functional and design requirement can be created. An example for each of these requirements is shown in the diagram pane in Fig. 3. The shown functional requirement (number 1 in Fig. 3) specifies the flight scenarios considered to determine the wear. The states hovering as well as vertical and horizontal flight are specified in a state machine diagram (number 2 in Fig. 3) connected to the formalization attribute of the functional requirement. The description of the flight modes in the functional requirements serve as a basis for the definition of the operating states with their associated loads and environmental conditions, which must be considered in the validation of the helicopter gearbox lifetime [27].

The design requirement (number 3 in Fig. 3), which is also shown, specifies the minimum maintenance interval at which the wear of the helicopter's main gearbox is allowed to exceed the permissible level. By connecting the design requirement with the SysML dependency *satisfy* with the parameter *time between overhauls* of the helicopter main gearbox (number 4 in Fig. 3) the requirement can be verified automatically during the simulation of the system architecture, even if changes in the specified boundary conditions occur.

Formalizing helicopter requirements in a requirements model with the motego profile offers advantages over document-based requirements capturing. The expectations of the stakeholders concerning the system become unambiguous and verifiable (cf. [8]). When requirements are linked to the other elements of the system model in the next development steps, changes in requirements directly influence all relevant models (cf. [5]). Consequently, the formalization of requirements with the motego profile has significant advantages in contrast to document-based and informal requirements.

Table 3 Elements of motego solutions layer needed for helicopter gearbox wear analysis

System Solution and Solution Elements	*Objective: Solution Concepts*
Solution concepts are specified in the motego method with system solutions and solution elements. These elements refine functions by a parametric description of contacts and structures geometry and material, as well as domain models and workflows for functional testing [3, 6, 7]	The architecture developed in the function layer already provides model input parameters. In order to integrate further parameters depending on domain-specific solution into the structure, solution concepts must be described
Physical Effect and Active Surface Set	*Objective: Contact Description*
On the elementary level, solution concepts are described by a physical effect which is used for the transformation of incoming into outgoing functional flows and a set of active surfaces between which the effect occurs [3, 5, 28]	Solutions only define a kind of container for specifying solution concepts. To ensure standardized parametric description of the contacts solution concepts, additional elements are required
Structure Set	*Objective: Structure Description*
On system level, solution concepts are described by subordinate solution elements and structure element set [10]. Structure elements define active surfaces realization within one physical structure	On system level solution concepts, i.e. structures with the associated contact surfaces, need to be described in the architecture
Domain Models	*Objective: Simulation Models*
In the motego method, simulation models are integrated into the system solutions and solution elements as so-called domain models [7]	To determine the contact wear directly in the architecture, external simulation models have to be integrated
Workflows	*Objective: Execution Sequences*
In the motego method, workflows define the execution of domain models for design, validation or optimization tasks and whether these are executed sequentially, parallel or iteratively [29]	The integrated and interconnected domain models must be executed in a defined sequence. These sequences need to be specified in the systemic wear model
Solution Model	*Objective: Directory*
A solution model serves as a kind of folder where all elements of the solution layer, such as solution elements and system solutions, as well as related tables, and diagrams are stored	To collect all elements of the solution layer for the helicopter gearbox as well as the tables and diagrams in which they are presented a kind of directory is needed

4.2 Functions

In order to develop a systemic wear model by combining individual simulation models, which is easy to understand and specifies the dependencies between the input and output parameters of the individual models, a structure is necessary. In the motego method, the function layer provides this structure. Table 2 lists the objectives to be satisfied by this structure in the context of systemic wear prediction in helicopter gearboxes, and the motego method elements that ensure this.

In conclusion of Table 2 the function layer of the DSL for the motego method needs to include the elements *elementary function, functional architecture, function model*, as well as *energy, material* and *signal flows*. Figure 4 illustrates the impact of embedding these elements in the motego profile on modeling helicopter gearbox architecture in CSM. Additional to the requirements model explained in Fig. 3, the function model can be used to implement the function layer of the motego method as part of the architecture. As elements of this layer, only function flows and functions can be created due to the context-specific constraints implemented in the motego profile (cf. containment tree on the left side of Fig. 4).

Besides the restrictions regarding structuring the elements in the containment tree, the motego profile also specifies connection rules between the elements. These context-specific restrictions are illustrated in Fig. 4 using all elements which can be modeled as part of a functional architecture—function flows which are transformed by a function, as well as hierarchical connections between a function and subordinate functions (see number 4 in Fig. 4).

To restrict the syntax between functions and function flows, further elements, so-called function flow ports, are implemented in the motego profile (cf. number 2 in Fig. 4). For these ports it is specified firstly that they can only be created as part of functions (cf. create element window in Fig. 4) and secondly that the port type is always a function flow. Thus, it is guaranteed that the function flows cannot be assigned to other elements than functions in the architecture of the motego method as well as the interfaces of the function cannot be specified by other elements than function flows.

An analogous approach is used to specify the dependency between functional architectures and their subordinate functions. To implement the constraint for this dependency, the architecture and element property are used as a further element of the motego profile. The architecture property enables the linking of the functional architecture with another subordinate functional architecture, whereas the element property enables the linking of subordinate elementary functions. As an example, the assignment of the elementary function “InDecrease Torque With Lubrication” to the functional architecture “Derive Secondary Forces” is presented in Fig. 4 (see number 3).

Functional modeling of the helicopter with the motegoProfile specifies the architecture solution-neutrally with well-defined interfaces. This solution-neutral structure with

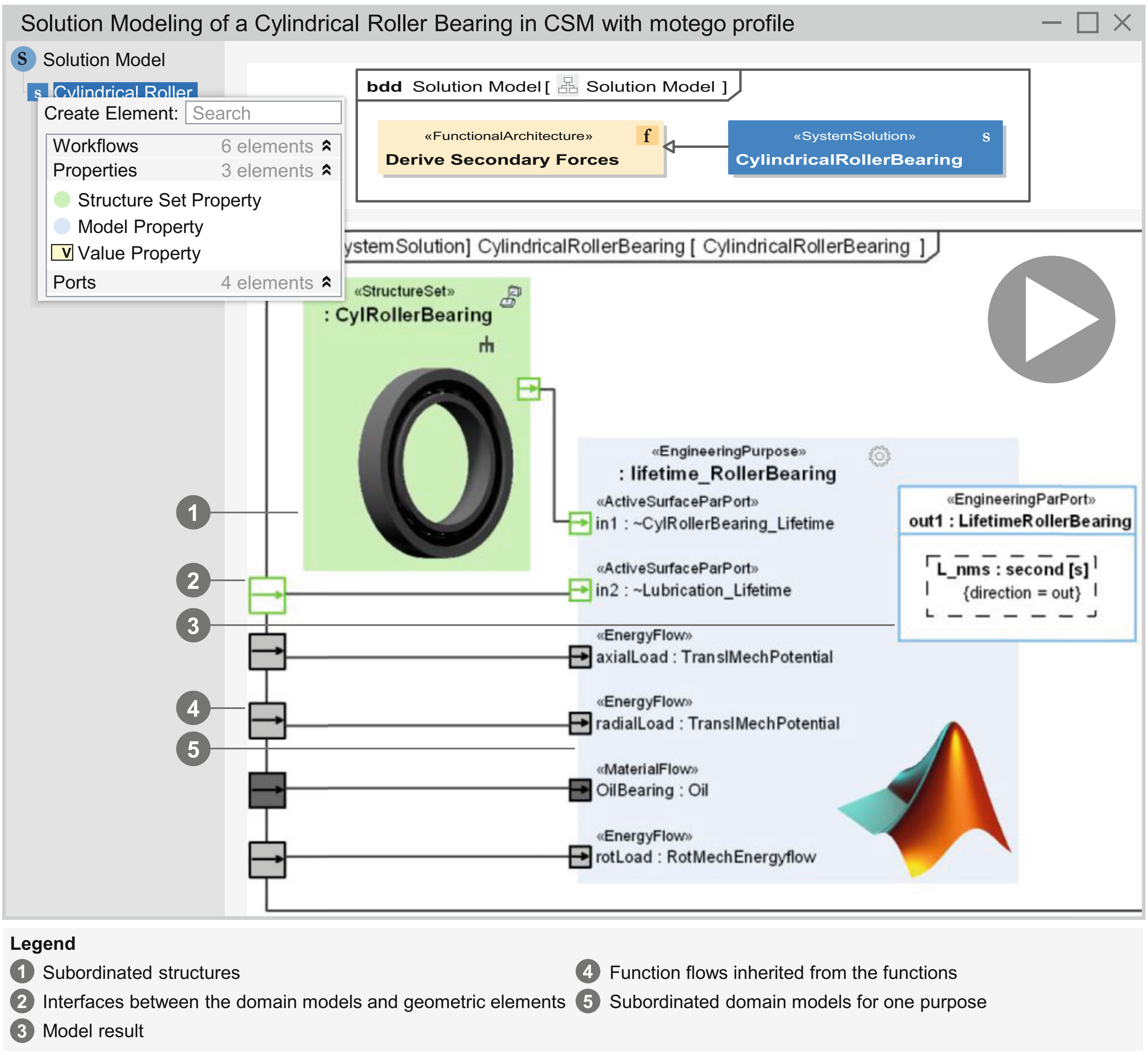
Solution Modeling of a Cylindrical Roller Bearing in CSM with motego profile
Solution Model
Cylindrical Roller
Create Element: Search
Workflows 6 elements
Properties 3 elements
Structure Set Property
Model Property
Value Property
Ports 4 elements
bdd Solution Model [Solution Model]
«FunctionalArchitecture» Derive Secondary Forces
«SystemSolution» CylindricalRollerBearing
ystemSolution] CylindricalRollerBearing [CylindricalRollerBearing]
«StructureSet» : CylRollerBearing
«EngineeringPurpose» : lifetime_RollerBearing
«ActiveSurfaceParPort» in1 : ~CylRollerBearing_Lifetime
«ActiveSurfaceParPort» in2 : ~Lubrication_Lifetime
«EnergyFlow» axialLoad : TranslMechPotential
«EnergyFlow» radialLoad : TranslMechPotential
«MaterialFlow» OilBearing : Oil
«EnergyFlow» rotLoad : RotMechEnergyflow
«EngineeringParPort» out1 : LifetimeRollerBearing
L_nms : second [s] {direction = out}
1
2
3
4
5
Legend
1 Subordinated structures
2 Interfaces between the domain models and geometric elements
3 Model result
4 Function flows inherited from the functions
5 Subordinated domain models for one purpose

Fig. 5 Solution modeling of a cylindrical roller bearings structure elements and a DIN ISO 281 [30] lifetime calculation model in CSM with motego profile

linked functional flows offers a significant advantage over current document-based approaches in which the models are directly associated with specific solutions, since not only one design variant can be tested, but also different solution concepts with their associated wear models can be efficiently exchanged and compared with each other. This advantage is particularly effective if the current design of the gearbox is analyzed with regard to wear and optimized on the basis of the results.

4.3 Solutions

Primary objective in the development of a systemic wear model of the helicopter gearbox is the integration of simulation models in the architecture across different hierarchical levels. To achieve this goal, specifications of the gearbox design have to be included in the architecture. Conceptual, parametric description of the design are specified in the motego method within solution layer. Table 3 lists the elements that must be included into solution layer to represent the systemic wear model with the DSL for the motego method in a software environment.

Table 4 Elements of motego product layer needed for helicopter gearbox wear analysis

Module	*Objective: Reorganizing Structures*
In the motego method modules are used to assembly structure elements in different views like mechanical design or production	For the product-centric reorganization of structure elements an element is needed, which represents this new grouping
Product Model	*Objective: Directory*
A product model serves as a kind of folder where all elements of the product layer, i.e. structure elements and modules, as well as related tables, and diagrams are stored	To collect all elements of the product layer for the helicopter gearbox as well as the tables and diagrams in which they are presented a kind of directory is needed

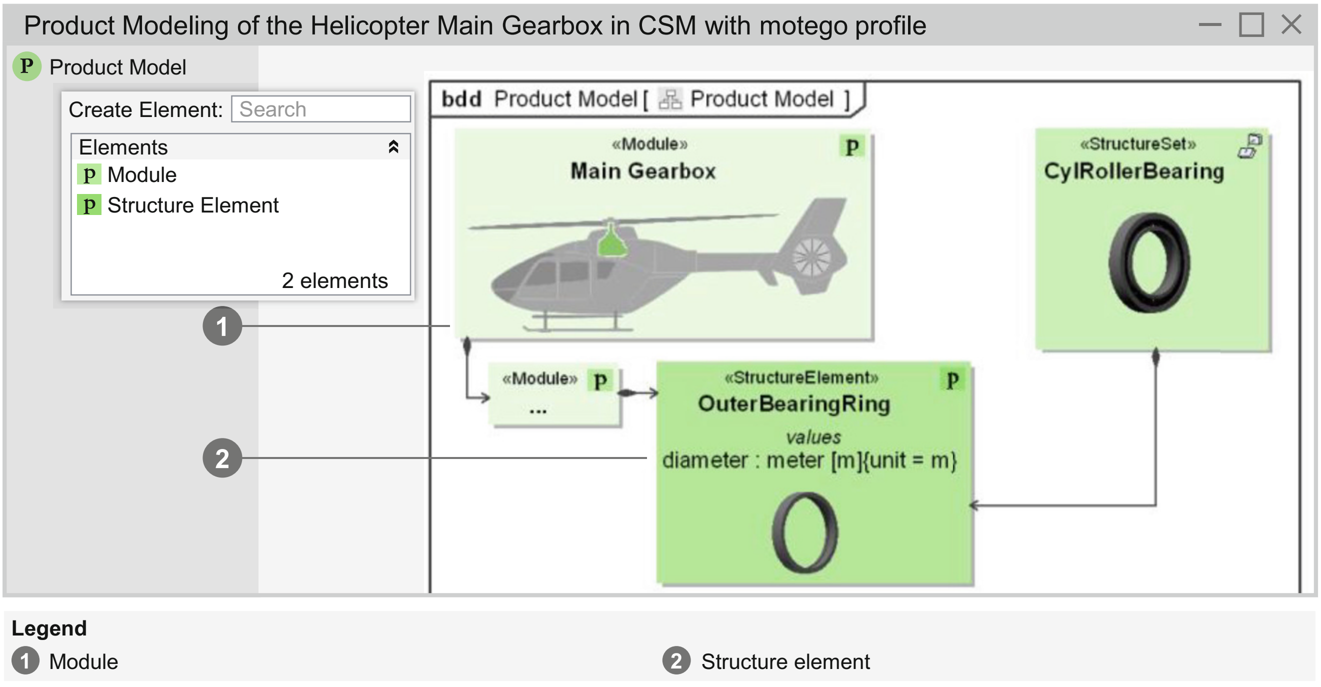

Fig. 6 Product modeling of the helicopter main gearbox in CSM with motego profile

Summarizing Table 3, the following elements must be implemented in the motego profile to build a systemic wear model—*system solutions, solution elements, active surface sets, structure sets, domain models*, and *workflows*. Figure 5 shows the application of these elements implemented in the motego profile using the example of a cylindrical roller bearing system solution that realizes the functional architecture "Derive Secondary Forces" shown in Fig. 4.

The upper part of the diagram panel in Fig. 5 shows the connection between the "Derive Secondary Forces" function shown in Fig. 4 and the realizing system solution of a cylindrical roller bearing. By associating the functional architecture with the system solution via SysML generalization relationship, the function flows defined in function are inherited by the system solution (number 4 Fig. 5). These functional flows serve as direct input for the lifetime calculation and thus in the validation of the system solution. Extension of the system solution with further elements is predefined by the motego profile. The elements that can be created as part of the system solution are restricted to workflows, as well as particular properties and ports. The particular properties specify the relationship of the system solution to domain models and structure sets. As an example of these relationships, Fig. 5 shows the structure set describing the balls, inner ring and outer ring of the bearing (number 1) as well as a simulation model for the determination of the bearing lifetime, which was implemented on the basis of DIN ISO 281 [30] (number 5). The lifetime model requires, in addition to the geometry and material properties of the bearing, input parameters from the inherited functional flows, such as the oil viscosity and the radial and axial loads, as well as geometry properties of other system solutions in the architecture, such as the type of lubrication system (number 2). By integrating the lifetime model into the system solution and linking the parameter inputs, the bearing lifetime can be automatically determined by simulating the architecture of the gearbox (number 3). The calculated lifetime can then be verified against requirements and transferred to further simulation models. An example of the loopback between the domain models' simulation results and the requirements is shown in Fig. 3 for the parameter "time between overhauls". This parameter is indirectly related to the bearing lifetime, as the lifetime must be greater than the selected maintenance intervals.

Modeling the solution layer of the motego method is a core aspect of the architecture when developing a systemic wear prediction model. As a result of implementing the solution model benefits of formalizing requirements and function layers become effective. A main advantage of the presented profile is the support of the alignment between the goals derived from requirements and properties of existing simulation models. Using the modeling method for model networks presented in [6], targets can be modeled as outputs of the purposes (such as "L_nms" in Fig. 5). These explicitly modeled targets can be aligned with the model signatures presented in [22]. By means of the created model networks and the formalized requirements, desired product functionalities can be tested and optimized. The parameterized consistent linking of the requirement, function, and solution layers also facilitates fast reaction to changes in requirements and the efficient adaptation of solutions.

4.4 Product

In the analysis and optimization of mechanical products such as the helicopter gearbox, especially mechanical engineers are involved. Since mechanical engineers are used to thinking in terms of components and modules, reorganizing the function-oriented architecture and results of solution layers becomes particularly relevant for better understanding. For this purpose, the product layer of the motego method offers good mechanisms since results from the solution layer can be presented in a reduced, comprehensive form. Since structure elements of the solution layer define which active surfaces of the solution elements are realized together in a physical structure, they serve as a bridge between the functionally developed active surfaces and the product realization. For this reason, they have to be included in the product layer. However, further elements are needed for the reorganization of the structure elements in a comprehensive, product-centric structure. These additional elements of the product layer are listed in Table 4.

Summarizing, the elements *module* and *product model* have to be implemented additionally in the motego profile for the product layer. Figure 6 shows a small section of the structure element reorganization in the helicopter gearbox product model. Figure 6 shows this structure element reorganization using the example of the outer bearing ring. The structure element "OuterBearingRing" is on the one hand part of the structure set "CylRollerBearing" shown in Fig. 5 and provides input parameters for the lifetime calculation (e.g. diameter). On the other hand, the structure element is part of the "Main Gearbox" module in the product layer. Via the structure elements, a close meshing of the functional product view in the solution layer and the geometry-oriented view in the product layer is created.

Similar to the other layers, the creation of elements in the product model is guided by the motego profile in such a way that only those elements and connections can be created that are defined in the architecture of the motego method. For the product model, this means that only modules (Fig. 6 number 1) and structure elements (Fig. 6 number 2) can be created in the product model and these are connected hierarchically via composition relationships.

The presented profile for the motego product layer represents the current state of the method. Interactions on this level, such as total cost calculations or design space compatibility, can currently be represented with SysML elements in the product model. Explicit stereotypes for a better description of interactions on the product layer will be developed in future. However, the presented motego profile already support the product development from requirements to product. Further advantages arise from the different views available within product layer, information such as production costs from other departments involved in the realization of the product, such as production planning, can be seamlessly integrated into the development process (cf. [6]).

5 Conclusion and outlook

Objective of this paper is the presentation of a DSL specially developed for the motego method, which enables the reasonable, standardized application of the method for the development of a systemic helicopter gearbox wear model. Therefore, relevant elements of the motego method for helicopter gearbox architecture development were identified and implemented as a DSL in a profile for the CSM. To illustrate advantages of the motego method DSL, excerpts from the helicopter systemic model development using the motego profile in the CSM have been presented in Fig. 3 to Fig. 6. Concluding, the motego profile significantly enhances the understanding and differentiation of the modeled elements and guides the developer through the architecture design process. This modeling guidance supports the engineer considerably in representing systems architecture in CSM in a method-compliant manner. Besides improved user experience, this has the advantage that architectures are always developed with a standardized structure and are therefore compatible with each other. As an example, the presented architecture of the bearing can be reused in other systems as well, such as in an electric powertrain.

The presented motego profile currently contain essential elements for the development or analysis of mechatronic products like the helicopter gearbox with the motego method and are freely available for the Cameo Systems Modeler 19.0 [23]. As next steps, extensions for further applications such as the development of medical devices are reasonable. In this application, the use of MBSE methods such as motego offers considerable potential for increasing the efficiency of development and certification processes, which could be exploited through the further development of the motego profile. Furthermore, modular extension of the motego profile for other methods such as variant or process management might be considered in order to support development processes holistically.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Alur R (2015) Principles of cyber-physical systems. MIT Press, Cambridge, Massachusetts, London
2. Eigner M (2021) System Lifecycle Management: Digitalisierung des Engineering. Springer Vieweg, Berlin, Heidelberg
3. Jacobs G, Konrad C, Berroth J et al (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, pp 243–263
4. (2023) MATLAB—MathWorks. https://de.mathworks.com/products/matlab.html. Accessed 2 Jan 2023
5. Zerwas T, Jacobs G, Spütz K et al (2021) Mechanical concept development using principle solution models. Iop Conf Ser Mater Sci Eng 1097:12001. https://doi.org/10.1088/1757-899X/1097/1/012001
6. Spütz K, Jacobs G, Konrad C et al (2021) Integration of production and cost models in model-based product development. JSS 09:53–64. https://doi.org/10.4236/jss.2021.912004
7. Spütz K, Berges J, Jacobs G et al (2022) Classification of simulation models for the model-based design of plastic-metal hybrid joints. Procedia CIRP 109:37–42. https://doi.org/10.1016/j.procir.2022.05.211
8. Höpfner G, Jacobs G, Zerwas T et al (2021) Model-based design workflows for cyber-physical systems applied to an electric-mechanical coolant pump. IOP Conf Ser Mater Sci Eng 1097:12004. https://doi.org/10.1088/1757-899X/1097/1/012004
9. Berges J, Spütz K, Jacobs G et al (2022) Automated identification of valid model networks using model-based systems engineering. Systems. https://doi.org/10.3390/systems10060250
10. Wyrwich C, Jacobs G, Spütz K et al (2023) Seamless function-oriented mechanical system architectures and models. (under review)
11. Friedenthal S, Moore A, Steiner R (2015) A practical guide to SysML: the systems modeling language, 3rd edn. Elsevier; Morgan Kaufmann, Amsterdam, Waltham
12. Object Management Group OMG (2019) SysML: OMG system modeling language. https://www.omg.org/spec/SysML/. Accessed 19 July 2022
13. Object Management Group (2017) OMG® Unified Modeling Language® (OMG UML®): Version 2.5.1
14. Hause M (2006) The SysML modelling language. In: Fifteenth European systems engineering conference, vol 9, pp 1–12
15. Moeser G, Grundel M, Weilkiens T et al (2016) Modellbasierter mechanischer Konzeptentwurf: Ergebnisse des FAS4M-Projektes. In: Tag des Systems Engineering, pp 419–428
16. Grundel M, Abulawi J, Moeser G et al (2014) FAS4M-No more: 'Please mind the gap! In: Tag des Systems Engineering
17. Weilkiens T (2016) SYSMOD—the systems modeling toolbox: Pragmatic MBSE with SysML, 2nd edn. Fredesdorf. version 4.1. MBSE4U booklet series. MBSE4U
18. Pohl K (2012) Model-based engineering of embedded systems: the SPES 2020 methodology, 1st edn. Springer, Berlin, Heidelberg
19. Oose Innovative Informatik eG, Em Engineering Methods AG, Karlsruher Institut für Technologie et al Mechanics Modeling Language (MechML): Version 1.0. https://fasform.de/content/ergebnisse/. Accessed 2 Jan 2023
20. Weilkiens T (2007) Systems engineering profile—SYSMOD. In: Weilkiens T (ed) Systems engineering with SysML/UML: modeling, analysis, design. Morgan Kaufmann; Safari Books Online, Burlington, Boston, pp 271–284
21. Regnat N, Gupta R, Jansen N et al (2022) Implementation of the SpesML workbench in magicdraw. https://doi.org/10.18420/modellierung2022ws-008
22. Zerwas T, Jacobs G, Kowalski J et al (2022) Model signatures for the integration of simulation models into system models. Systems 10:199. https://doi.org/10.3390/systems10060199
23. Dassault Systèmes (2022) Cameo systems modeler. https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/. Accessed 5 Oct 2022
24. No Magic Product Documentation (2022) Using requirement terms glossary. https://docs.nomagic.com/display/CRMP190/Using+Requirement+terms+glossary. Accessed 24 Oct 2022
25. Drave I, Hillemacher S, Greifenberg T et al (2019) SMArDT modeling for automotive software testing. Softw Pract Exper 49:301–328. https://doi.org/10.1002/spe.2650
26. Rumpe B (2012) Agile Modellierung Mit UML: Codegenerierung, Testfälle, Refactoring, 2nd edn. Xpert. press Ser.. Springer, Heidelberg, Berlin
27. Zerwas T, Jacobs G, Brand L et al (2023) Customer-centric and function-oriented development of mechatronic systems. DSEC Conference in "Forschung im Ingenieurwesen"
28. Koller R (1998) Konstruktionslehre für den Maschinenbau: Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen. Springer eBook Collection Computer Science and Engineering. Springer, Berlin, Heidelberg
29. Berges J, Spütz K, Zhang Y et al (2023) Reusable workflows for virtual testing of multidisciplinary products in system models. DSEC Conference in "Forschung im Ingenieurwesen"
30. DIN ISO 281:2010-10, Wälzlager – Dynamische Tragzahlen und nominelle Lebensdauer (ISO_281:2007)