CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-022-00631-0

ORIGINAL PAPER

Integrated model-based design and functional hazard assessment with SysML on the example of a shock control bump system

M. Schäfer[1] · A. Berres[1] · O. Bertram[1]

Received: 22 February 2022 / Revised: 15 July 2022 / Accepted: 5 December 2022

Abstract
Integrating new functions into the aircraft can, for example, increase performance or reduce fuel consumption. Since the installation of such additional functions increases the overall aircraft complexity, it is crucial to adapt methods and tools that support the development and ensure traceability, consistency, and verifiability. In this context, model-based systems engineering and the associated Systems Modeling Language (SysML) have been established as a standard methodology. This paper presents an overview of a system development and modeling process with SysML at the concept design stage using a position-variable shock control bumps system as an example. In addition to the system modeling, safety and reliability analyses have to be considered during the design process. To keep both, the model and the associated safety assessment consistent, this work introduces an extension of SysML to enable the execution of a functional hazard assessment (FHA) according to the ARP4754A and ARP 4761 guidelines. This is the first step in conducting a model-based safety assessment. Furthermore, a modeling process with concepts management methods is performed. In summary, the presented modeling process consists of three main parts: the system modeling, functional hazard assessment and concept management.

Keywords SysML · FHA · MBSE · MBSA · Model-based design

1 Introduction

The design of complex systems requires a methodological approach which contains different steps such as problem and requirements formulation, solutions search, documentation and evaluation [1]. Different design methods were developed to support the system design process [2, 3]. In the aerospace field, especially the INCOSE systems engineering process is well known and frequently used. This holistic interdisciplinary approach describes the system design over the complete lifecycle [4]. The process is originally document-based. However, it is difficult to reuse such document-based developments and models [5].

Through the introduction of the system modeling language SysML, model-based systems engineering (MBSE) became more and more important for the design of complex systems [6]. The modeling with SysML can enable a consistent and reusable system design. Besides, through the representation of different views, MBSE creates the opportunity to improve collaboration among different engineering fields [5]. It should be noted, that SysML is only a modeling language and not a methodology or design process. Different software tools are using SysML to enable a computational design of systems, like Papyrus, Enterprise Architect or Cameo Systems Modeler.

An important part of the design process is the conceptional design phase. In this phase, about 70% of the cost for the system development are set and the solution space is defined [7–9]. Because of that, changes in later stages are costly and require a high effort. Since this phase of development is so important, it is astonishing that there is a gap in the methodology and approach in the field of MBSE. In a survey within the INCOSE Model-Based Conceptual Design Working Group, it was determined which issues often occur in the concept design phase among systems engineers [10]. It was found that 40% of the difficulties mentioned are exacerbated by the application of MBSE in the concept design phase. The main reasons given were the lack of best practice methods and stopping criteria for modeling within model-based design. The lack of methods for model-based conceptual design is also reinforced by the fact that variant

✉ M. Schäfer
Michael.Schaefer@dlr.de

[1] Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Brunswick, Germany

Published online: 19 December 2022

modeling, which is important in the concept phase, is only possible to a limited extent in SysML, as the language offers no specific elements for this task [5, 11].

In the development of new aircraft systems, safety analysis plays an important role alongside the design of different concepts. Especially in the case of safety–critical functions, these aspects must be included in the development and design at an early stage. For certification of large passenger aircraft in Europe, EASA CS 25.1309 [12] is applied. The SAE ARP4761 [13] and ARP4754A [14] (Aerospace Recommended Practice of the SAE) show methods to meet the EASA requirements for system certification. In 2020 the OMG published the SysML Profile Risk Analysis and Assessment Modeling Language (RAAML) [15]. Profiles such as RAAML can be used to extend the UML and SysML with safety-specific elements. Unfortunately, in this safety profile FHA is only partly covered. However, in the preliminary design stage, FHA is required to identify safety–critical functions early and considering them during design. Therefore, FHA on the aircraft as well as on the system level is mandatory for a successful and traceable development of safety–critical systems. There are approaches for the model-based creation of an FHA [16], but not an integrated approach within the SysML modeling language.

This paper describes an integrated approach to the conceptual design of an aircraft system, introducing the design method and a way to consider concept variants. Existing methods are presented, adapted, and combined into an overall approach. The main focus of the work is the development of the system within one integrated model-based process. For this purpose, an FHA profile in SysML was furthermore developed. The profile is based on the methods from ARP4761 and ARP4754A. This profile can close the gap between system engineering and the RAAML profile, enabling continuous system engineering and safety analysis within SysML. In addition, some tool-specific features were integrated to facilitate the work with the profile. The approach is demonstrated by a use case study on a shock control bump system.

2 System design with SysML

For the model-based design in a preliminary design phase, a process consisting of different existing, adapted, and new implemented methods was created. Figure 1 summarizes the preliminary design process of a system within the MBSE environment (blue) in a simplified representation integrating the safety assessment (red). In addition to the main steps of the model-based development (e.g. functional analysis, architecture design), the most important interfaces involved with the model are also shown. The focus of this paper is on the main steps marked in green:

- Functional Analysis
- Execution of the FHA
- Documentation of different concepts

Accordingly, this chapter is separated into three parts, describing the basic approach to system modeling, followed by an overview of FHA profile development and its implementation in SysML, and concluding with methods for modeling various concepts.

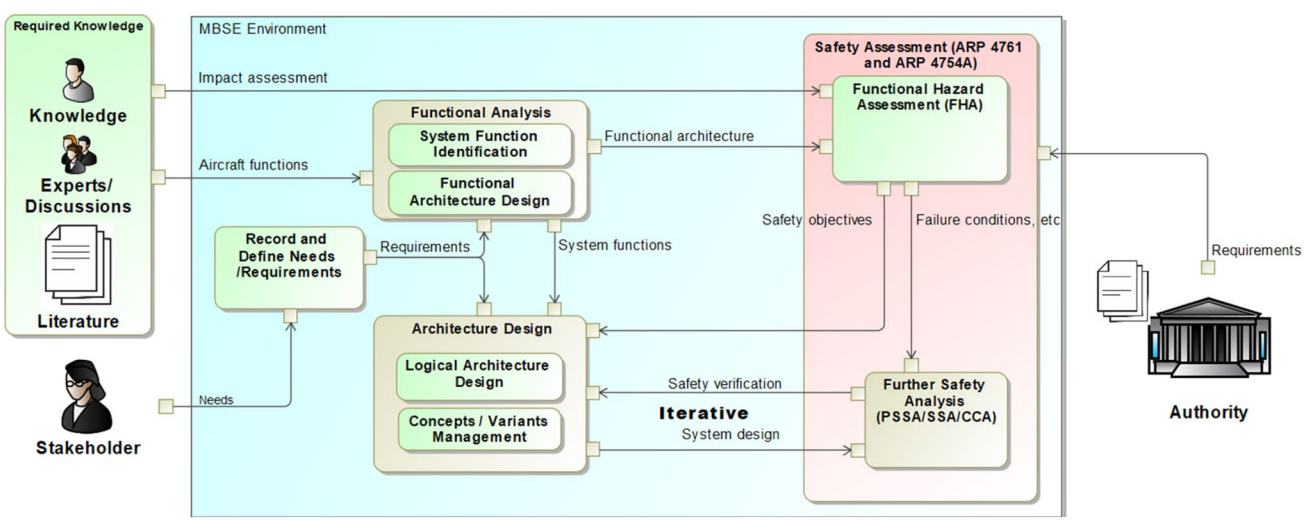

Fig. 1 Integrated model-based design (blue) and safety analysis process (red) in one MBSE environment and the interfaces to the surrounding stakeholders

2.1 System design methodologies

There are several methodologies for model-based systems engineering to support the designer through the development process. Among the best known are:

- Harmony [17]
- Rational Unified Process for Systems Engineering (RUP SE) [18]
- INCOSE MBSE [19]

Harmony and INCOSE MBSE are aligned with the V-model development process [20], whereas the RUP SE methodology is divided into four main model levels: Context, Analysis, Design, and Implementation. This structure is derived from the Zachman style matrix [21]. It can be noted that the method is only rarely referenced in current scientific publications and has therefore probably decreased in relevance. In addition, the disadvantage of many of these methods is that they are very abstract or derived from software development and therefore difficult to apply to the development of mechatronic systems. Based on this issue, more technical-orientated methodologies were developed such as Software Platform for Embedded Systems (SPES) [22] or the MagicGrid Framework [23]. SPES provides a language and tool-independent methodology (e.g. activities, techniques) for system modeling. SpesML is a combination of the SPES Methodology and SysML. This combination should therefore enable easier application in practice. MagicGrid Framework is a methodology, which is based on the SysML, but is tool independent. A detailed description of the methodology is given in the next section.

2.1.1 MagicGrid framework

The MagicGrid Framework has an approach similar to the RUP SE and can be applied by any SysML-tool. Further advantages of the method are the focus on technical systems and detailed documentation [23]. This makes it easier to get started with model-based system design. Therefore, this method was chosen as the basis for the system design process. Furthermore, the diagrams of MagicGrid are extended with elements from other methodologies to improve readability (see also Sect. 4).

The following paragraph is a brief overview of the parts of the MagicGrid framework used in this work. A detailed description of the framework can be found in [23]. Figure 2 illustrates the main modeling and analysis steps that have to be executed during the system development according to the MagicGrid framework. The structure of the framework follows the standard ISO/IEC/IEEE 15288 [24]. The standard describes the processes for the development of systems over

			PILLAR			
			Requirements	Behavior	Structure	Parameters
DOMAIN	Problem	Black Box	Stakeholder Needs	Use Cases	System Context	Measurements of Effectiveness (MoE)
		White Box		Functional Analysis	Logical Subsystem	MoE for Subsystems
	Solution		System Requirements	System Behavior	System Structure	System Parameters
			Subsystem Requirements	Subsystem Behavior	Subsystem Structure	Subsystem Parameters
			…	…	…	…
			Component Requirements	Component Behavior	Component Structure	Component Parameters
	Implementation		Physical Requirements			

Fig. 2 MagicGrid framework with the corresponding problem domain (green) and the solution domain/implementation (blue) based on [23]

the entire life cycle. The framework is based on the technical processes described in the standard.

As shown in the figure, the methodology is divided into a problem (green), solution, and implementation domain (blue). The approach in this work is to use the problem domain. This domain is separated into a black box and a white box. The columns separate the methodology into the 'four pillars of SysML' [6] All defined diagrams within SysML can be divided into these four pillars (requirements, behavior, structure, parameters). Therefore, the division into the individual columns already gives an indication of the diagrams to be used in the respective sections. The procedure of the modeling takes place domain-wise, starting from the top to the bottom.

At the beginning of the methodology, the black box subdomain is addressed. The black box analysis is intended to identify the inputs and outputs of the system and to set the boundaries towards other systems. The internal behavior and structure are not considered at this point of the analysis. In the first step, the stakeholder needs are collected in a requirements table. In the next step, the system context is captured and the use cases are defined. These processes are closely linked since the delimitation between the system of interest (SoI) and other systems takes place first. SoI describes the system under consideration at a certain hierarchical level of a global system. Within the system context, the system boundaries are defined and the interfaces with other systems from the environment are identified. The system context is usually defined in an internal block diagram. Then, use-case scenarios are created using use-case elements. Based on the use cases, the interaction between the system of interest and external systems can be specified in more detail. The external system can be, for example, an environment, another technical system or a person. The analysis of the use cases in connection with the external system takes place in a use case diagram. The use case definition also includes a more detailed specification of the use case scenario. This description can be done, for example, using activity diagrams and

action elements. In the last step of the black box domain, the non-functional stakeholder needs are identified and modeled as measurements of effectiveness (MoE) in a block definition diagram.

Initially, in the white box problem domain, the main functions of the system are identified from the detailed use case descriptions which were defined in the black box domain. Based on these functions, a functional analysis of the SoI is performed. For this purpose, the function is separated into sub-functions. The connections between these sub-functions are modeled in an activity diagram. Subsequently, logical components are allocated to the sub-functions. Considering the function analysis, a logical architecture, and their interfaces can be modeled in an internal block diagram. The next step corresponds with the last step from the black box domain. However, the assignment of further MoE to individual sub-systems can be omitted in case no further MoEs are required. Finally, the sub-functions from the functional analysis and all MoE are allocated to the stakeholder needs in a matrix diagram.

2.1.2 General amendments for MagicGrid

MagicGrid is a comprehensive method that allows to model technical systems in SysML. However, there are some aspects that can be considered additionally, especially at the beginning of the modeling process. These aspects can facilitate further activities. Therefore, an additional phase before starting the actual problem domain is proposed in this work. This here called '0 Superior Analysis' shall mainly serve to formulate a system objective and to improve the communication and documentation within the modeling and system development.

At the beginning of development and modeling, the main objective of the system should be defined and documented. This formulation should serve in the model to clearly define the focus during development. For this purpose, the stereotype'Objective' is introduced. The stereotype is based on the required elements of the SysML and also has an ID and text attribute. Accordingly, the objectives are created in a requirements diagram. A further aspect introduced in this analysis phase is the allocation of the system main function or functions into the overall system. For this purpose, depending on the system knowledge, a function hierarchy is built up in the top-down or bottom-up principle, with the overall system on top of the hierarchy and one or more functions of the SoI on the bottom. This analysis can support the definition of use cases, system context, and also safety analysis. The modeling can be performed in a block definition diagram. Figure 3 shows an example of a functional hierarchy.

The previous amendments mainly concern the starting phase of development and modeling. Another addition is related to the representation of the stakeholder needs and requirements. These are not stakeholders in the sense of project management, but rather sources of requirements. For this purpose, the stereotype 'ReqSource' was created. The representation of these stakeholders should be created at the beginning of the project and maintained as well as supplemented during the project, if necessary. The allocation of the 'ReqSource' and stakeholder needs to take place afterwards in the problem and solution domain. This documentation facilitates cooperation and can be helpful in the course of the project to find suitable contact persons. Figure 4 shows the final package structure of the model after the problem analysis phase. Likewise, the glossary for abbreviations and explanations of terms can be placed in this package.

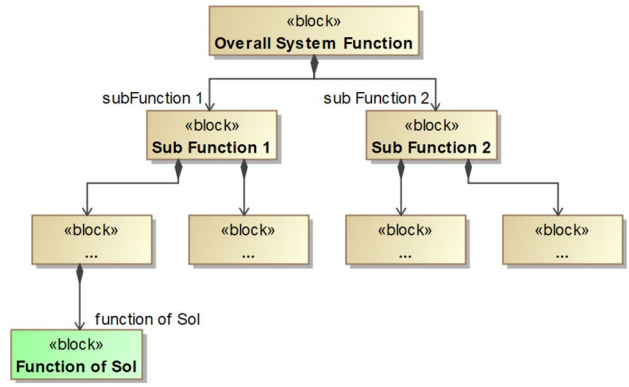

Fig. 3 Decomposition of generic system functions with the identified SoI (green)

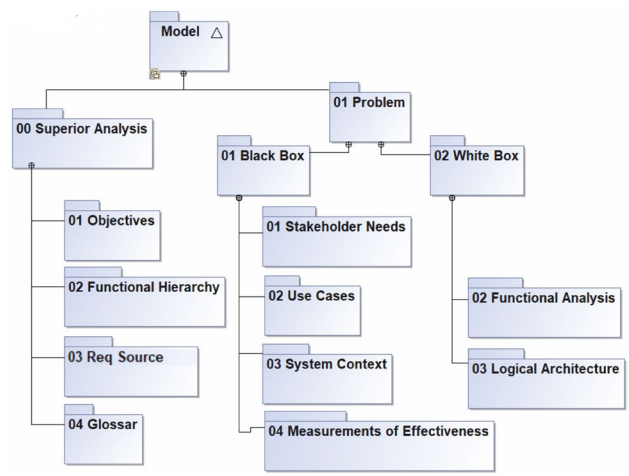

Fig. 4 New problem domain model structure with the new'0 Superior Analysis' part

2.2 Functional hazard analysis

In order for new technologies and systems to be integrated into aircraft, they have to be approved by the responsible authorities. Important aspects of the approval process are the safety requirements for the systems. Without proof of the required reliability, the approval of new systems for aircraft is not possible. Furthermore, the identification of safety requirements' violation at a late development stage may lead to major changes which result in high costs and time efforts. Therefore, safety aspects must be addressed at an early stage of system design.

The ARP 4761 corresponds to a standard for the development of safety–critical aviation systems. Part of the ARP is the Functional Hazard Assessment which is intended for hazard analysis in early development phases. The FHA describes a process that includes eight steps (Fig. 5). Since the development of safety–critical systems requires the close cooperation of system and safety developers, these two domains must be considered. To facilitate this collaboration, ISO 15288 for systems engineering was also considered. The result of the analysis is a generalized process, which should enable a simple and lean implementation of the FHA in four phases (see also [16]).

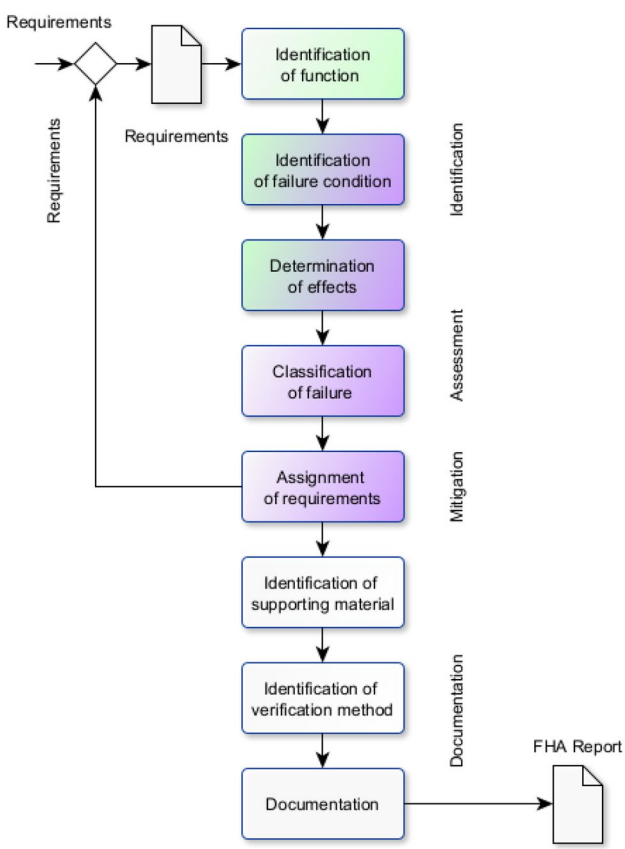

Fig. 5 FHA process with main steps

The task of the FHA is to identify and assess hazards that may result from a system. If necessary, strategies that can minimize the existing hazards risk should be identified. Thus, FHA provides the basis for risk control of hazards emanating from a system. Risk control can be enabled if the following four phases (identification, assessment, mitigation, documentation) are performed in the FHA. However, before the FHA can start, a system description must be provided. This serves to understand the functions to be implemented as well as their interactions. In addition, this description is used for communication between the domains.

2.2.1 Identification of hazards

During the identification phase, a systematic approach is used to identify potential hazards. In the standards given above, the use of guide words is recommended for this purpose. These are intended to point out certain typical hazards in a system during the analysis. For example, the guide word "late" can indicate that a sensor value is provided too late. This missing value can lead to an incorrect control command and the loss of a system. But the use of guide words has the disadvantage that the thinking of the participants is specifically steered in one direction. To counteract this effect, creative methods like brainstorming can be used. From our observations, it has been shown that this can identify hazards that are not yet known, but the quality of the results depends heavily on the participants and their expertise.

2.2.2 Analysis and assessment of the hazards

Once the hazards have been identified, the risk for each must be analyzed and evaluated. For the analysis, the safety expert can draw on his domain knowledge and the knowledge of the system developers. In this context, the knowledge can be provided by ontologies and applied with machine learning methods as shown in [25]. Another method to provide an ontological interpretation of hazard is described in [26]. Hazard tracking and knowledge management systems can be used to support the analysis. These databases collect knowledge from accidents that have already occurred. The analysis results indicate the resulting damage, name possible causes as well as underlying hazards of functions and systems. In addition, these databases may contain information about the operation of the systems. Based on the information collected, the severity of the occurrence of each hazard is assessed. This so-called classification represents the basis for the reliability requirements of the corresponding function. After all, hazards have been assessed and the results documented, this assessment should be discussed with the system engineers.

There is a special challenge for systems based on new concepts or technologies. It is difficult to draw back on

existing knowledge for those systems and therefore, the hazards must be assessed to the best of our knowledge and belief. Therefore, it must be ensured that the use of simulations or the evaluation of test results of similar systems can prevent an overly optimistic assessment and enable a realistic assessment. This approach can ensure that novel systems can be operated with acceptable risks. Through the subsequent operation of these systems, the missing knowledge can be continuously reduced by gathering experience.

2.2.3 Hazards mitigation and derivation of requirements

In the final step, hazards that can lead to unacceptable damage must be minimized. These damages may include irreversible destruction of resources or the environment or personal injury. The result of hazard minimization is a list of possible actions or requirements for the system. This list includes architectural, design, or implementation suggestions. These suggestions should be based on verifiable ideas or accumulated experience.

2.2.4 Documentation

In the analyzed standards, documentation is required as the final step of the analysis. These documents can be part of a certification of a system. This step cannot be generalized due to different certification processes. However, it is possible to continuously document the analysis results during the individual steps. Either Word, PDF or HTML documents can be generated automatically. The automatic generation of documents can also be done from model-based tools. The continuously generated documentation can then be reused for the creation of the certification documents.

2.2.5 A SysML profile for FHA

To be able to perform a model-based FHA in SysML, a profile extending the model-based language is needed. In the following paragraph, the developed profile is described, which is currently comprised of four packages. These packages are system description, core, library and views as shown in Fig. 6 and are described in detail later in this section.

A stereotype 'SystemFunction' was defined in the system description package. The meta model of the element is shown in Fig. 7. This stereotype serves as a basis for the analysis and is used to declare blocks or classes in block definition diagrams (bdd) and internal block diagrams (ibd) as a function. This way a functional analysis can easily be done as usual, using the bdd for decomposition and identification of the system functions. Moreover, the relationships between the system functions can be modeled in an ibd.

The core package provides the essential stereotypes of the profile to execute the model-based safety assessment:

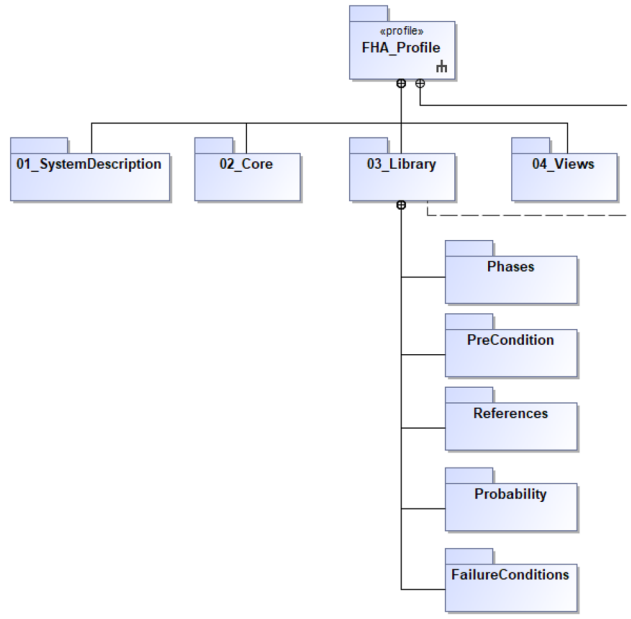

Fig. 6 Package structure of the tool independent part of the FHA profile

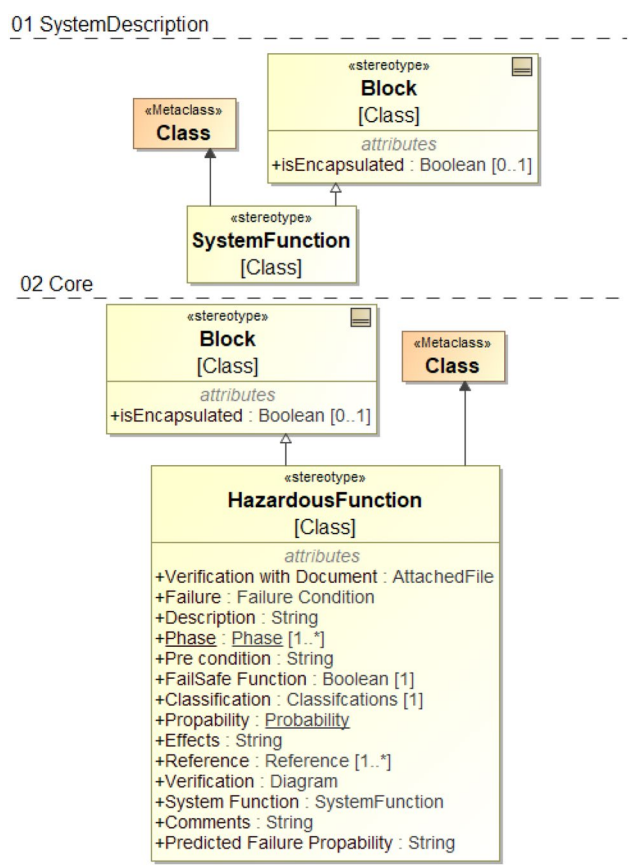

Fig. 7 Extract of the meta model of SystemDescription package with the 'SystemFunction' element and Core package with the 'HazardousFunction' container element

'HazardousFunction', 'FailureCondition', 'Reference', and 'ReferenceType'. The 'HazardousFunction' stereotype is the main element of the model-based FHA. This element is a container that groups all relevant information for the analysis of a system function, such as 'Phase', 'FailureCondition', and so on (Fig. 7). The 'FailureCondition' is defined according to the ARP4761 as 'a condition with an effect on the aircraft and its occupants, both direct and consequential, caused or contributed to by one or more failures, considering relevant adverse operation or environmental conditions'. 'Phase' is required during system definition for modeling system operation and later for hazard identification. To provide material that supports the arguments for an assessment, 'Reference', and 'ReferenceType' can be specified.

The package library contains specific model elements which are predefined and separated into the categories 'FailureConditions', 'Phases', 'PreConditions', 'Probability', and 'References'. 'FailureConditions' describe a failure of a function in an abstract way (e.g. degradation of function, partial loss of function, etc.). Depending on the development phase of the system, these abstract elements can be used directly or they can be specified in more detail. In the 'Phases'-package, for example, the relevant flight and maintenance phases were predefined. These phases can later be used or extended by the safety engineer as well. In the preconditions package, the stereotype 'AnyCondition' is defined. Derived from this, 'EnvironmentalCondition' such as fog and rain have been pre-modeled. Besides, in 'Probability', qualitative safety objectives were defined according to the ARP4761, as well as a wide range of quantitative objectives.

The views package contains the tables for the FHA and the references. The FHA table shows the content of the 'HazardousFunction' elements. This table corresponds to the representation of the FHA table in the ARP4761. In the reference table the author, source type, date, and source, a document for example, are displayed.

2.2.6 Customizations in SysML-tool

Most of the common SysML-tools offer the ability to extend the functionality beyond the SysML-standard. NoMagic's Cameo Systems Modeler for example offers the ability to add additional functionality to a profile and to extend the tool through an open application programming interface (API). These modifications can enable an efficient work in projects. For this reason, the profile has been extended with some specific functions, which are shown in Fig. 8.

An additional element offered by the Cameo System Modeler is the so-called smart package. With these packages, the content is automatically generated according to previously defined rules. Therefore, smart packages can be used for automated logical structuring of model elements. Here, two

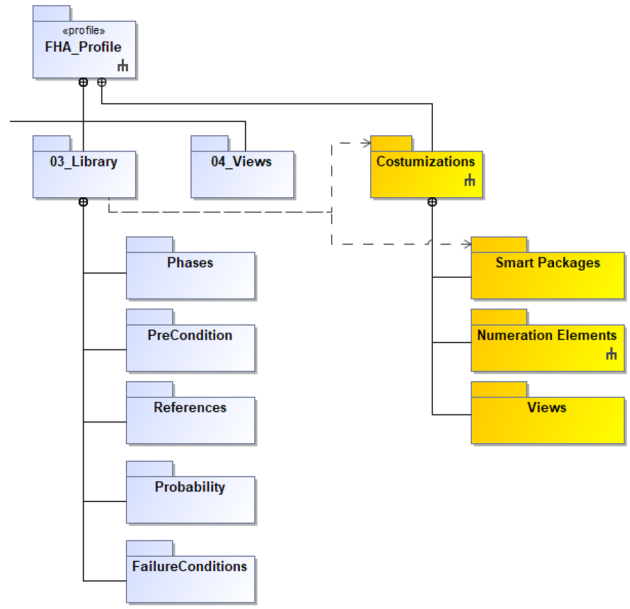

Fig. 8 Package structure of the customizations (yellow) of the FHA profile

smart packages with two different rules were implemented. The smart package with the stereotype 'UnusedFunctions' displays all system functions that have not yet been used in the functional safety hazard assessment table. Another smart package was implemented to display the system functions, which are classified as high risk (hazardous or catastrophic).

In the package 'Numeration Elements', rules have been created to automatically provide each 'HazardousFunction' element with a sequential number. These functions are used to assign a specific label and number to each hazardous function element in the FHA table.

2.3 Concept modeling

In the early design stage concept creation is an important task to find a proper solution. For the development of concepts, there are many methods in the field of systems engineering that are intended to span a large solution space and support the search for suitable solutions. Holliger-Uebersax describes in [27] general methods and principles for solving problems using morphological methods. Bardenhagen and Rakov developed an advanced approach specifically for aerospace applications [28]. The theory of inventive problem solving (TIPS/TRIZ) [29] is another method for finding solutions to problems. It describes practical methods and tools with a large range of applications. The level of detail of the concepts often depends on the stage of development of the system. A methodical approach makes sense for new systems and for the further development of existing systems. This is often followed by a rough pre-selection of the

most promising concepts. There are many different methods available for the evaluation of concepts (e.g. morphological approach [30], weighted sum model [31], genetic algorithms [32], trade-off analysis [33]).

After a pre-selection, there are often several concepts on the shortlist that cannot be fully evaluated at an early stage of development. However, these concepts should be documented and included in the development process to be able to carry out a further evaluation at a given design stage.

2.3.1 Variation modeling with feature trees

There are various methods for the modeling of concepts. A large part of these methods is more or less based on feature-oriented domain analysis (FODA) [34]. The core of the FODA method is the feature tree. In this model, all possible features are shown in one representation. In this case, features can be equated with possible variations. Predefined symbols in the feature tree can be used to define whether a feature is optional or required. Figure 9 shows an example of such a feature tree.

According to the illustration, the symbols describe three possible relationships between the features and the system. The optional and mandatory branches are self-explanatory. In the recent past, however, a filled circle is often added at the end of the connection of a mandatory feature. Furthermore, the 'alternative' connection is often divided into the OR (filled) and the XOR (non-filled). Thus, with the XOR (= exclusive OR), only one of the features can be used for a variant. With this representation method, large quantities of variants can be represented in a compact model. This is also one of the reasons why feature modeling can be found in many concepts of variant modeling methods.

The approach in [35] called Variant Modeling with SysML (VAMOS) also uses feature modeling to represent variants. Unlike other variant methods, the VAMOS approach is designed specifically for SysML. The basic structure of the method is shown in Fig. 10. As can be seen in the illustration, the model is separated into three areas. The core (green) represents the model itself. In this core, variation points, which represent model parts that can be

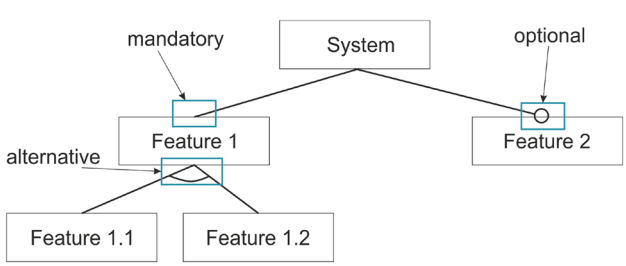

Fig. 9 Generic example of a Feature tree with the three possible relationships (mandatory, alternative, optional) based on [34]

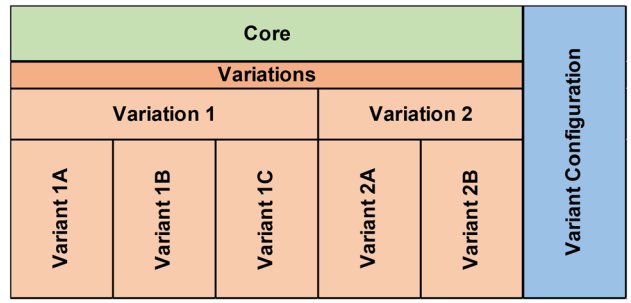

Fig. 10 VAMOS structure based on [35]

varied, are defined. The variation area (red) represents the feature tree in this method. Here, as in FODA, features and their relationship to each other are defined and connected to the corresponding variation points from the core. In the variant configuration (blue) the individual variations are defined.

In addition, there is also Product Line Engineering, which is mainly used in the automotive industry to manage variants. This method can also be implemented as Model-Based Product Line Engineering (MBPLE) in SysML ([32, 36–38]). According to [39] the MBPLE can be separated into four steps. Figure 11 illustrates this process.

In the first step of the MBPLE, a feature tree is created as a diagram in SysML. The root is modeled with the 'rootfeaturegroup'-element. Variations or features are modeled with the stereotype 'featuregroup'. In addition, enumerations can be used to allow the selection of variations. The relationships are modeled using multiplicity instead of symbols as used in FODA. Once all variation points are modeled, instances from the feature tree are defined. These instances represent the individual variants or concepts. In the next step, or in parallel, the so-called 150% model is created. This model contains all elements, both variable elements and elements that must be present in all variants. Then, the variable elements have to be connected to the variation points from the feature tree via a 'feature impact' relationship. The creation of the 150% model is not limited to the structure representation. The relationship to existing or non-existing elements can be created in any diagram. Thus, the 150% model can contain multiple views. The final step of the MBPLE is the Model to Model transformation (M2M-transformation), starting from the 150% model. With the M2M-transformation, individual variants are created from the 150% model and the instances of these variants. In this step, all elements that were not assigned to the respective instance variant are removed from the model. Depending on the tool used, this step can be automated.

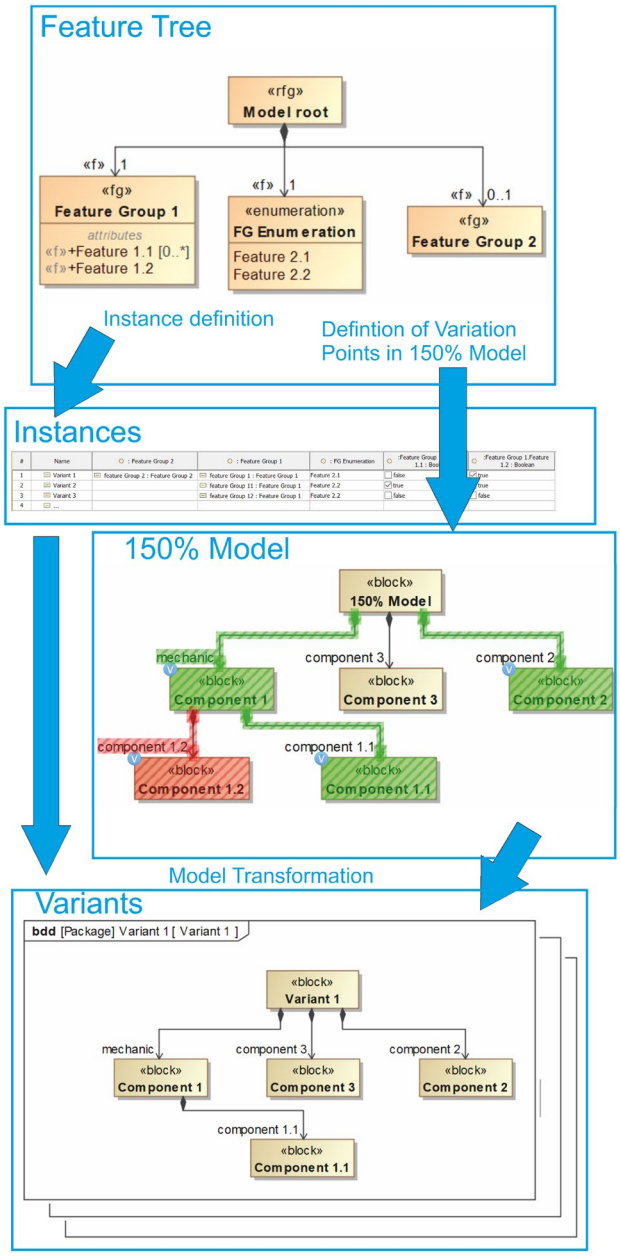

Fig. 11 Four steps of MBPLE based on [39]: Create feature tree, define instances, create 150% Model, and generate variants with M2M-transformation

2.3.2 Application in the design approach

The MagicGrid framework described above also considers the creation of several concepts. The solution domain (see Fig. 2) is run through again for each individual concept. Therefore, the method seems unsuitable for modeling many concepts. For the approach described in this work, the MBPLE method is used and combined with aspects from the VAMOS method. MBPLE is applicable with all common SysML tools in combination with additional extensions (e.g. pure-systems [40], BigLever Software [41]). Furthermore, the methodology is already partially integrated into some SysML tools (e.g. Cameo Systems Modeler [39]).

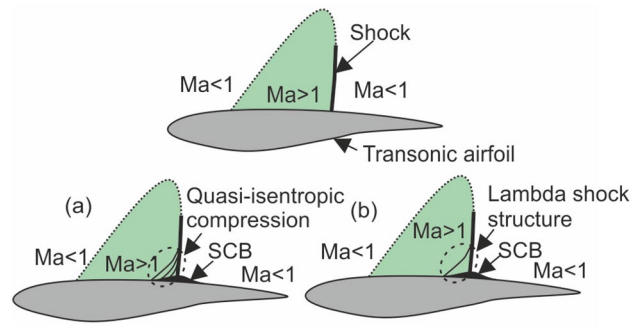

Fig. 12 SCB functionality with a quasi-isentropic compression **a** based on [42] and lambda shock structure **b** based on [43]

3 SCB-system as a case study

The approach for the conceptional design process described in this paper was developed as part of the Federal Aeronautical Research Program (LuFo) in the project Move-IntegR. In this project, design solutions for multifunctional control surfaces at the trailing edge of laminar wings are developed. This includes the design of a multifunctional shock control bump (SCB). The SCB is a bump on the upper side of the wing that can positively influence the flow of transonic airfoils. The bump is positioned in the area of the compression shock to reduce its negative effects. Figure 12 shows the simplified functionality of an SCB and is explained below.

In the project, two use cases for the SCB are investigated. Firstly, the use of the bump in the transonic drag rise regime. In this use case, the wave drag which results from the compression shock can be reduced during the flight. Depending on the SCB contour, two different flow effects can occur (Fig. 11a and b). The SCB can lead to a quasi-isentropic compression with many weak shocks (a). Alternatively, a so-called lambda shock structure can appear (b). Both types can reduce the wave drag of the wing. Another potential application of an SCB is the buffet-onset delay. By correctly positioning a bump with a suitable shape, the flight envelope can be extended by stabilizing the shock position. The shape, height, and position of the bump must be adjusted depending on the flight condition and the application, to gain the greatest possible advantage from the technology.

The wing airfoil has a high impact on the shock position variation. Especially with turbulent airfoils, the shock position variation is sensitive towards flight conditions. The development of a positionally fixed shock control bump for a laminar wing with an unswept leading edge was investigated

in a predecessor project of Move-IntegR [44]. In Move-IntegR, a position variable SCB is developed for a laminar wing with swept leading edge, as it is expected to provide greater advantages for the aircraft. Initial investigations in the project have shown that the shock position at the outer part of the wing is in the area of the spoilers. Accordingly, the SCB-system must be considered in relation to the entire spoiler system.

4 Modeling of shock control bump

The integration of the shock control bump into the spoiler system represents a multi-disciplinary challenge for various sub-disciplines. In addition to the aerodynamic design, the structure, and system must ensure that implementation is possible both technically and in terms of safety and reliability. For the implementation of such a complex and novel system, model-based systems engineering was chosen as the approach. The model is built and maintained by the systems engineer and serves to support the traceability of the requirements, system development, safety analysis, and documentation. In addition, individual views can be used to improve communication among the individual sub-disciplines. For this purpose, the methods presented in chapter 2 are applied to support the design of the SCB-system at an early development stage.

For the system design and implementation of methods, the SysML-tool Cameo Systems Modeler is used. Besides the standard SysML language, this tool has features that facilitate the system modeling and therefore, are used within the design process. However, most of the methods presented here, can also be implemented in other SysML tools, because the basis are mostly standard SysML/UML elements.

4.1 Modeling of the SCB-system

The modeling of the problem domain is based on the methods described in Sect. 2.1. At the beginning, the objectives for the SCB are defined and system functions are decomposed. The decomposition reveals the two main functions of the SCB-system:

- Drag reduction
- Buffet onset control

As described in the procedure, the most important requirements stakeholders were documented. In the next step, the black box was modeled according to the MagicGrid framework. As an example, the use case diagram of the SCB-system is shown in Fig. 13.

The illustration shows that the use cases are divided into two parts. The lower part shows the use cases of the spoiler

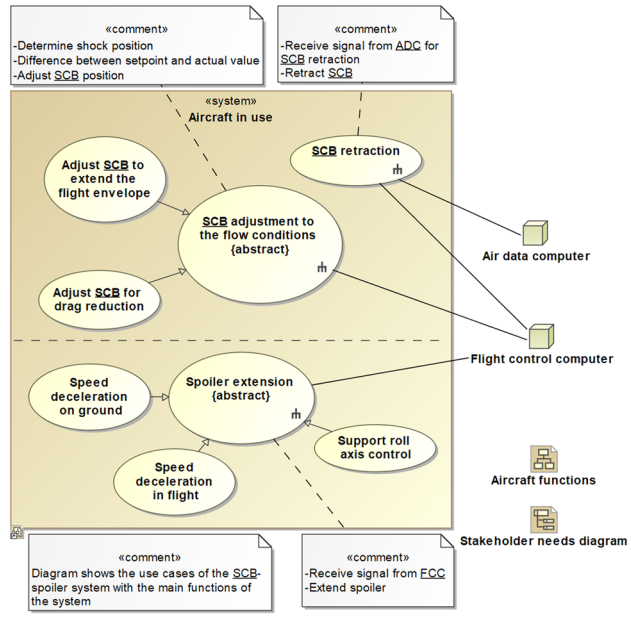

Fig. 13 Use case diagram of SCB-spoiler-system separated into SCB use cases (top) and spoiler use cases (bottom)

functions. These have to be taken into consideration since the SCB system is part of the spoiler actuation system. The upper part of the figure shows the SCB-system use cases. An abstract use case is split into the two main use cases of the SCB system: adjust SCB for drag reduction and adjust SCB to extend the flight envelope. The SCB system interacts with the flight control computer and the air data computer. In addition to the standard representation of the use case, the method from another methodology (SYSMOD [45]) was adapted to present the basic steps of the individual use cases in the comments. Furthermore, a short description of the diagram was added, as well as links to the corresponding diagrams. This approach increases the readability of the individual views, the navigation within the model, and therefore, is applied to almost all diagrams within the model.

4.2 FHA for shock control bump functions

After the main functions of the system have been identified, an initial safety analysis of the system can be carried out using the developed FHA profile. The main goal of the FHA is to identify safety–critical functions, mitigate risks, and derive safety requirements during development. Therefore, the FHA should be updated again and again during the course of development and expanded if necessary.

The two main functions of the SCB system are each defined as system functions with the profile exclusive stereotype ‘SystemFunction’. Then, the possible failures for these two functions are identified and analyzed. Therefore, basic failure types are provided by the library integrated into the

profile (e.g. total loss of function, incorrect function). The appropriate types can be used in the model to create the specific failures of the SCB-system. The identified failures of the drag reduction function are shown in Fig. 14.

Next, the FHA table is created and the previous information is assigned to a functional hazard element. The failures must now be examined for the effect they have on the overall system. This investigation is an important point in the creation of the FHA, as the effects can only be reliably estimated with suitable knowledge. Especially in the development of new systems, a lack of knowledge to provide a conclusive assessment often exists. The assessments must therefore be constantly reconducted during the course of development. This is also the case with the SCB-system. On the basis of expert panels consisting of aerodynamics, structural, and systems engineers, initial assessments of the effects were made. Based on the effects, a classification can then be made according to CS 25.1309. This in turn leads to the safety objective for the respective failure. Figure 15 shows an extraction from the FHA table of the SCB-system for the two main functions.

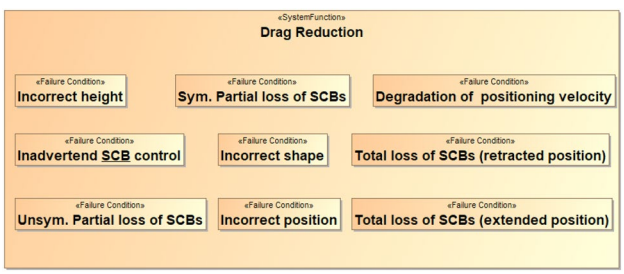

Fig. 14 Identified failures of the Drag Reduction Function of the SCB-system

4.3 Variant modeling

In the project, more than 30 basic concepts were identified from literature research, preliminary work [46] or created. Then, these concepts were evaluated by experts in panel discussions. In the process, eight concepts emerged as promising possible solutions. Due to the early development stage, the emerged concepts cannot be finally evaluated because of the limited knowledge on the system. Therefore, these concepts are modeled into the SysML model using MBPLE to allow further evaluation at a more advanced development stage.

The various features of the concepts were modeled in a feature tree (Fig. 16), according to the MBPLE. Since the system is in an early stage of development, the modeling was performed at a high level of abstraction. For example, no distinction was made between different piston actuator technologies, and mechanisms were described abstractly without further specification. In next, the stances of the concepts were defined. In addition, the 150% model structure view was created. The individual feature elements were mapped to the respective model elements from the 150% model. Besides the structural view, other diagrams where modeled, e.g. a parameter diagram to determine the mass of the system. The advantage of this procedure is reflected in the next step. In the M2M transformation an instance is selected. From this, the respective model of the concept with all associated views is created. All elements that do not belong to the concept are removed from the model. Figure 17 shows an example of such a model. In addition to the model, a sketch with notes on the concept was added to the structure view. This procedure originates from the VAMOS methodology and is especially helpful in enabling the representation of concepts to be more comprehensible.

#	Hazard No.	System Function	Phase	Failure	Effects	Classification	Propability
1	FH-1	Drag Reduction	Cruise Landing Descent	Unsym. Partial loss of SCBs	Assymmetric drag increase	Hazardous	<1.0E-5
2	FH-2	Drag Reduction	Cruise	Incorrect position	Increased drag + fuel consumption	Minor	<1.0E-3
3	FH-3	Drag Reduction	Cruise Descent Landing	Total loss of SCBs (extended position)	Increased drag + fuel consumption	Hazardous	<1.0E-7
4	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
12	FH-11	Buffet Onset Control	Cruise Climb Landing	Total loss of SCBs (extended position)	going around: higher resistance; increase in drag and fluel consumption, flow seperation	Hazardous	<1.0E-7

Fig. 15 Part of the SCB-system FHA-table with the main attributes

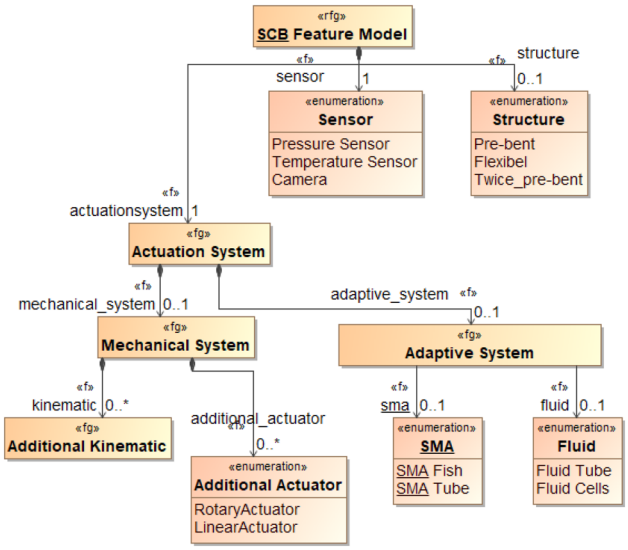

Fig. 16 Feature model of SCB-system features with all possible variable parts

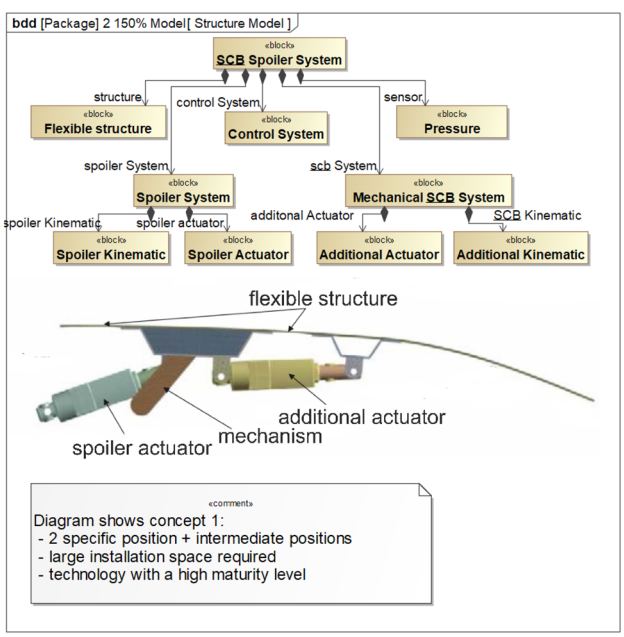

Fig. 17 Concept model after M2M-transformation of a two-actuator SCB-spoiler-system concept

5 Conclusion

In this work, methods enabling or supporting the modeling of systems in an early design stage in SysML were presented. Existing methods, such as the MagicGrid framework, VAMOS, and MBPLE, were applied, extended and combined. In addition, a new profile enabling FHAs to be carried out within SysML and thus representing a further step towards holistic development in SysML was shown. The application of the methods was demonstrated using the SCB-system. The modeling of the problem domain, including white and black box analysis, was shown as an example. Subsequently, an FHA was carried out with the help of the developed FHA profile. The main elements of the introduced FHA profile are based on the basic SysML elements of the OMG. This allows the profile to be adapted and integrated into all common SysML tools without major modification. In addition, current research is being conducted on the connection between the RAAML and FHA profile. These investigations are a further step in the progress towards a holistic safety analysis within the SysML. For this purpose, there is also contact with the OMG.

By integrating predefined elements and views, it should be ensured that the FHA profile does not require previous knowledge. In addition, documentation was written to support the work. In the next step, further adjustments will be made to the profile to improve the workflow. For example, an automated generation of the safety requirements from the FHA is being developed. Besides, validation functions will be integrated into the profile to ensure consistency and completeness.

In addition to the safety analysis, previously developed concepts were modeled within SysML using methods from variant management. The MBPLE method was primarily used for this purpose. During the application, however, it was found that the modeling of concepts with many different features can be time-consuming, as here the 150% model can become extensive. Another difficulty is the modeling of abstract concepts. A general understanding of the concept cannot often be ensured on the basis of the modeling. For this reason, the MBPLE was extended to include representations from VAMOS. This way, the readability of diagrams can be significantly increased.

However, it can be stated that the modeling of concepts is not yet possible to full satisfaction within SysML. Further development of the methods is therefore still necessary. In particular, the current concepts have to be further developed on a higher level of detail in the 150% model. This also includes the creation of additional views, such as mapping the internal structure in an ibd. In addition, the possibility of performing trade-off analyses using the MBPLE method to support concept selection has to be investigated.

Acknowledgements Funded by the Federal Republic of Germany. Funding source: Federal Ministry for Economic Affairs and Climate Action based on a resolution of the Bundestag. The project MoveIntegR (20W1729C) is part of the Federal Aeronautical Research Program (LuFo V-3).

Funding Open Access funding enabled and organized by Projekt DEAL. Bundesministerium für Wirtschaft und Klimaschutz, 20W1729C, Michael Schäfer.

Data availability The data that support the findings of this study are available from the corresponding author, upon reasonable request.

References

1. Brunetti, G., Golob, B.: A feature-based approach towards an integrated product model including conceptual design information. Comput. Aided Des. **32**, 877–887 (2000). https://doi.org/10.1016/S0010-4485(00)00076-2
2. Boehm, B. W.: Guidelines for verifying and validating software requirements and design specifications. In Euro IFIP 79, 711–719. North Holland (1979)
3. VDI: VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte (1993)
4. INCOSE: Systems Engineering Vision 2020. International council on systems engineering, technical operations (2007)
5. Woelkl, S. and Shea, K.: A computational product model for conceptual design using SysML. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 635–645 (2009). https://doi.org/10.1115/DETC2009-87239
6. OMG: OMG system modeling language. Version 1.6," Milford, CT, USA: Object Management Group, Inc (2019)
7. Walden, D. D., Roedler, G. J., Forsberg, K.: INCOSE Systems engineering handbook version 4: Updating the reference for practitioners. In INCOSE International Symposium (2015). ISBN: 978-1-118-99940-0
8. Li, T., Lockett, H., Lawson, C.: Using requirement-functional-logical-physical models to support early assembly process planning for complex aircraft systems integration. J. Manuf. Syst. **54**, 242–257 (2020). https://doi.org/10.1016/j.jmsy.2020.01.001
9. Boothroyd, G.: Product design for manufacture and assembly. Comput. Aided Des. **26**, 505–520 (1994). https://doi.org/10.1016/0010-4485(94)90082-5
10. Morris, B. A., Harvey, Robinson, K. P., Cook, S. C.: Issues in conceptual design and MBSE successes: insights from the model-based conceptual design surveys. In INCOSE International Symposium (2016)
11. Abulawi, J.: A SysML-based approach to exploring innovative system ideas for aeronautical applications. 31st ICAS, 817–826 (2018). ISBN: 978-1-5108-7501-2
12. EASA: CS-25 certification specifications and acceptable means of compliance for large aeroplanes (2016)
13. SAE: ARP 4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. Warrendale: PA, USA: Society of Automotive Engineers Inc (1996). https://doi.org/10.4271/ARP4761
14. SAE: ARP 4754A: Guidelines for development of civil aircraft and systems. Warrendale, PA, USA: Society of Automotive Engineers Inc (2010). https://doi.org/10.4271/ARP4754A
15. OMG: Risk analysis and assessment modeling language (RAAML). https://www.omg.org/spec/RAAML (2020). Accessed 20 June 2021
16. Berres, A. and Bittner, T.: A seamless functional hazard analysis for a fuel cell system supported by spreadsheets. ESREL - European Safety and Reliability Conference (2021). http://dx.doi.org/https://doi.org/10.3850/978-981-18-2016-8_114-cd
17. Hoffmann, H.-P.: Model-based systems engineering with Rational Rhapsody and Rational Harmony for systems engineering. Deskbook 4.1 - System Engineering Best Practices with Rational Solution for Systems and Software Engineering (2011)
18. Wahli, U., Irani, M., Magee, M., Negrello, A., Palma, C., Smith, J.: Rational business driven development for compliance. IBM Redbooks (2006)
19. Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML: The systems modeling language. Morgan Kaufmann, US (2014). (**ISBN: 0128002026**)
20. Graessler, I., Hentze, J., Bruckmann, T.: V-models for interdisciplinary systems engineering. Proceedings of the design. 15th International Design Conference, 747–756 (2018). https://doi.org/10.21278/idc.2018.0333
21. Zachman, J. A.: The concise definition of the zachman framework. https://www.zachman.com/about-the-zachman-framework (2008). Accessed 7 Aug 2021
22. Pohl, K., Hönninger, H., Achatz, R., Broy, M.: Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology. Springer, Heidelberg (2012). (**ISBN: 3642346138**)
23. Aleksandraviciene, A. and Morkevicius, A.: MagicGrid book of knowledge - A practical guide to systems modeling using MagicGrid from No Magic. Vitea Litera (2018)
24. ISO/IEC/IEEE: ISO/IEC/IEEE 15288: 2015 - Systems and software engineering - System life cycle processes (2015)
25. Daramola, O., Stålhane, T., Omoronyia, I., Sindre, G.: Using ontologies and machine learning for hazard identification and safety analysis. In: Managing Requirements Knowledge, pp. 117–141. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-34419-0_6
26. Zhou, Jiale, Hänninen, K., Lundqvist, K., Provenzano, L.: An ontological interpretation of the hazard concept for safety-critical systems. The 27th European Safety and Reliability Conference ESREL'17, 18–22 Jun 2017, Portoroz, Slovenia. (2017). https://doi.org/10.1201/9781315210469-157
27. Holliger-Uebersax, H.: Handbuch der allgemeinen Morphologie: elementare Prinzipien u Methoden zur Lösung kreativer Probleme. MIZ, Zürich (1980)
28. Bardenhagen, A., Rakov, D.: Advanced morphological approach in aerospace design during conceptual stage. Facta Univ. Ser. Mech. Eng. **17**, 321–332 (2019). https://doi.org/10.22190/FUME180110005B
29. VDI: VDI 4521 Inventive problem solving with TRIZ - fundamentals, terms and definitions (2016)
30. Rakov, D. and Bardenhagen, A.: Analysis and synthesis of aircraft configurations during conceptual design using an advanced morphological approach. In Deutscher Luft-und Raumfahrtkongress. Darmstadt (2019)
31. Zangemeister, C.: Nutzwertanalyse in der Systemtechnik: eine Methodik zur multidimensionalen Bewertung und Auswahl von Projektalternativen. BoD-Books on Demand (2014)
32. Arifin, H.H., Ong, H.K.R., Jie, D., Wu, D., Nasis, C.: Model-based product line engineering with genetic algorithms for automated component selection. In: CSDM Asia/CSDM. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73539-5_23
33. Berres, A.: Trade-off analysis for different architectures of safety-critical systems. ESREL - European Safety and Reliability Conference (2017)

34. Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., Peterson, A. S.: Feature-Oriented Domain Analysis (FODA) feasibility study. Carnegie-Mellon University Software Engineering Institute (1990)
35. Weilkiens, T.: Variant modeling with SysML. MBSE4U Booklet Series (2012). ISBN: 3981787579
36. Kang, K.C., Lee, J., Donohoe, P.: Feature-oriented product line engineering. IEEE Softw. **19**, 58–65 (2002). https://doi.org/10.1109/MS.2002.1020288
37. Ziadi, T., Jézéquel, J.-M.: Product line engineering with the UML: deriving products. In: Software Product Lines, pp. 557–586. Springer, Berlin (2006). https://doi.org/10.1007/978-3-540-33253-4_15
38. Hummell, J. and Hause, M.: Model-based product line engineering-enabling product families with variants. In 2015 IEEE Aerospace Conference (2015). http://dx.doi.org/https://doi.org/10.1109/AERO.2015.7119108
39. No Magic: Product Line Engineering Plugin. https://docs.nomagic.com/display/PLE190SP3/19.0+LTR+SP3+Version+News (2019). Accessed 7 June 2021
40. Beuche, D. and Papajewski, H.: pure-systems GmbH. https://www.pure-systems.com. Accessed 16 Jul 2021
41. Krueger, C.: BigLever Software, Inc. https://biglever.com. Accessed 2 Jul 2021
42. Birkemeyer, J., Rosemann, H., Stanewsky, E.: Shock control on a swept wing. Aerosp. Sci. Technol. **4**, 147–156 (2000). https://doi.org/10.1016/S1270-9638(00)00128-0
43. Ogawa, H., Babinsky, H., Pätzold, M., Lutz, T.: Shock-wave/boundary-layer interaction control using three-dimensional bumps for transonic wings. AIAA J. **46**, 1442–1452 (2008). https://doi.org/10.2514/1.32049
44. Werner, M.: Application of an adaptive shock control bump for drag reduction on a variable camber NLF wing. In 2018 AIAA Aerospace Sciences Meeting (2018). https://doi.org/10.2514/6.2018-0789
45. Weilkiens, T.: Systems Engineering with SysML/UML: Modeling, Analysis, Design. Elsevier, Netherlands (2011). (**ISBN: 8131222489**)
46. Künnecke, S.C., Vasista, S., Riemenschneider, J., Keimer, R., Kintscher, M.: Review of adaptive shock control systems. Appl. Sci. **11**, 817 (2021). https://doi.org/10.3390/app11020817