State-based safety analysis method for dynamic evaluation of failure effect

Yuqian Wu¹ · Gang Xiao¹ · Miao Wang¹

Received: 5 August 2020 / Revised: 22 October 2020 / Accepted: 3 November 2020 / Published online: 19 November 2020
© The Author(s) 2020

Abstract

System state that represents a combined influence of internal and external system parameters on the overall failure effect plays a significant role in failure effect analysis. The traditional safety analysis methods can hardly evaluate the overall failure impact due to the dynamic failure behaviors in diverse system interaction situations. To overcome this problem, this paper proposes a state-based safety analysis method for dynamic evaluation of the failure effect combining the situation factor. First, a hierarchical modeling framework that includes functional logic, physical architecture, and the failure mode is constructed, and then the cross-linking relationships between items are characterized by the state machines. Particularly, the event transmission mechanism and the global attribute updating mechanism are designed to realize the state synchronization of various systems, thus enabling the global propagation of failure. The feasibility of the proposed method is verified by simulations. The Enterprise Architect platform is used to model the aircraft integrated surveillance system and analyze the effects of different failure modes in typical situations. The proposed method complements the failure effect evaluation accuracy considering the dynamic interaction situations, thus realizing the global perception of the safety state, enhancing the dynamics and integrity of the failure effect analysis process.

Keywords Failure effect analysis · Model-based safety analysis (MBSA) · Model-based system engineering (MBSE) · System simulation

1 Introduction

Safety is a priority requirement in aircraft system design and operation. The failure effect analysis plays a significant role in the safety analysis. The traditional failure effect analysis methods, such as the Failure Mode and Effects Analysis, or the FMEA for short [5, 16], evaluate failure effects in the form of tables to verify the safety requirement. Despite the time and cost of a manual model construction [13], an increase in system complexity has posed the limitation on the accuracy of the failure effect analysis, which yields to the inability of the traditional static analysis method in the system reaction

prediction, including merge the dynamic influence of both internal and external parameters on the current system safety state. Therefore, the coherence between the system design model and the safety model should be further strengthened [2].

In recent years, the model-based safety analysis, or the MBSA for short [8, 12], has been used to combine the safety analysis process and design process. The existing MBSA methods, including the hierarchically performed hazard origin and propagation studies [11, 14], Rodon [9], and AltaRica [1], have proven the reliability of failure effect analyses utilizing formal system models. However, the full industrial implementation of those methods is still limited due to the high computational cost and insufficient understanding of the equation-based formal models in the background of complex system interactions.

The State Machine for Automation of Reliability-related Tasks using Information FLOWs (SmartIfFlow) represents another MBSA method that has been designed to automate the safety analysis process [10, 7, 3]. Compared to the formal safety analysis methods, this method possesses the

✉ Miao Wang
miaowang@sjtu.edu.cn

Yuqian Wu
wuyuqian@sjtu.edu.cn

Gang Xiao
xiaogang@sjtu.edu.cn

¹ School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China

advantages of dynamic behavior description and state transformation visualization. The state-based analysis methods have facilitated the dynamic evaluation of the failure effect, thus realizing the global perception of the safety state.

The application of the SmartIflow method to the complex systems remains limited, since the way of expressing various situations in the failure state modeling and updating process is still obscure, which limits the accuracy and comprehension of the failure-effect analysis results. Specifically, the relationship between the system state and current failure situations of the inner and external systems has become highly correlated with the increase in system interaction complexity. Therefore, the systematical safety modeling framework that can reflect the joint effect of various situations, namely, the comprehensive and dynamic updating mechanism of various states, should be explored.

Therefore, this paper proposes a state-based safety analysis method for a dynamic description of the failure effect, considering the situation of multiple system coherence. The proposed method includes the global state updating mechanism specially designed to conduct the dynamic analysis and the global perception of the safety state under various situations. The proposed method enhances the accuracy and comprehension of the failure effect analysis process, contributing to the integration and traceability of the model-based system engineering (MBSE) and MBSA method in the early development process of complicate systems such as avionics.

The rest of the paper is organized as follows. The state-based modeling framework for multi-layer and multi-system state relationships is presented in Sect. 2. In Sect. 3, the system state modeling and the dynamic expression methods of safety behavior are described. The simulation results of the aircraft integrated surveillance system are given in Sect. 4. In Sect. 5, the conclusions are drawn, and future research directions are presented.

2 Proposed modeling framework

2.1 Overall modeling framework

This section introduces the flight operation decomposition framework from the perspective of system engineering. The proposed modeling framework is illustrated in Fig. 1.

In the proposed framework, the flight consists of flight tasks assembly. The tasks are achieved through function combination. Similarly, the functional objective is realized through the subsystem functions combination, which is achieved by the physical components and public resources interaction. The safety behaviors considering failure modes are represented by state machines, which are embedded in

design models of each layer. Besides, state machines from different layers can interact with each other.

To achieve the safety analysis objective from the situation perspective, in the proposed framework, each system layer is modeled and presented in the form of structure and behavior diagrams. Specifically, the block definition diagram and internal block diagram are combined to depict the structural decomposition, and the state machine diagram is introduced to represent interactional behaviors of the different system modes, as well as mapping elements of different layers.

2.2 Aircraft level function modeling

Aircraft-level functions represent the means used to complete tasks through the chain form in a specific scenario, following the principle that aircraft-level functions are independent of each other. To structuralize the aircraft functions, this paper distributes these functions into different abstraction levels according to the application purpose. The functions in the same layer are connected by directional straight lines to illustrate logical interactions (refer to Fig. 2, while the functions of different layers are organized by decomposition lines constituting a solid diamond on the one side of the line. Each function block possesses attributes that constrain the function state and the applicable flight phase.

Then the logic equations of the aircraft-level task completion state are derived based on the functional chains. The state of each function is regarded as a global Boolean variable that corresponds to the conditions of the external models of the same layer and lower layers.

On this basis, each aircraft-level function is allocated to one or some particular systems for implementation according to the system interaction relationships (refer to Fig. 2 and functional objective of each system. The mapping relationships between functions and systems are illustrated in Fig. 3, where the connection stereotype between functions and systems is represented as <aggregation to whole>.

2.3 System structure modeling

In the system-level model, each system entity is defined by the block definition diagram (BDD) and a series of internal block diagrams (IBDs). Horizontal and vertical decompositions of the system are performed using the BDD according to the functions and physical architecture of the system, as illustrated in Fig. 4. As for the horizontal decomposition, the aircraft functions are expanded to system-level functions and subsystem-level functions, following the functional architecture relationships established in the IBDs. The vertical decomposition is a layer-by-layer decomposition of the system's physical structure, ranging from the subsystem level to the hardware and software levels.

Fig. 1 Modeling framework depiction

Specifically, each model of a given abstraction level is associated with an IBD, describing the interfaces, attributes, operations, and constraints of a particular block as well as the relationships with other blocks. The components of the IBD transmit information with each other via data ports, as shown in Fig. 5.

2.4 System behavior modeling

Behavior modeling follows the hierarchical principle that combines the top-down modeling strategy with the bottom-up behavior interaction strategy.

To specify behaviors of entities, each model from the component to the system block is attached with a state machine diagram to describe the state-based behaviors of that block and its interactions with external entities.

The principles of behavior modeling are as follows:

- Each behavior performed to accomplish a specific function must be specified and scheduled either in the form of an operation in the state block or a trigger in the state transition, which is able to be triggered by specific events.

Key behaviors related to the function completion include the input parameter processing procedure, key parameter calculating procedure, information transmission procedure, control and display procedure, and so forth.

- Each exchangeable message between the entity and external entities should be represented either by a trigger in the state transition, an attribute in the state block, or embedded in an operation in the state block.

The messages to be modeled include the required input parameters to complete the certain function such as the altitude parameter for the terrain threat judgment, as well as the

Fig. 2 Aircraft level function chains (left) & System interaction relationships (right)

external information which will influence the function state, such as the failure of the physical component related to the function.

The example of the state-based behavior model containing the behaviors and messages is shown in Fig. 6, which also depicts the relationship between the structural model and the behavior model. The details of behavior modeling based on the state machine diagram are discussed in detail in Sect. 3.

2.5 Model relationship establishment

After creating the elements of each layer, the relationships between the elements in the same layer and different layers are normalized, as shown in Table 1.

On the left side in Fig. 7, the elements' relationships represent the decomposition structure of the models of different layers. On the right side of the framework, the behavior model of each layer is matched with the corresponding model of the left side. The behavior models of different layers are connected by <dependency>.

3 System safety situation modeling

To realize the combination of the safety state synchronization of various systems with dynamic safety behavior expression, the system safety situation is introduced to describe the global safety state.

Fig. 3 Aircraft level function and system mapping relationships

Fig. 4 System structure decomposition by the BDD

Definition 1: Safety situation represents the background of various system interactions through multiple current activated states, that enables the dynamic network of information sharing and state synchronization between different systems from the global perspective.

Via the situation, the behaviors inside the same modeling layer, as well as the behaviors of the models of different layers, are intertwined, thus enabling dynamic evaluation of the safety state.

Particularly, situations are expressed by the state machines assembly embedded in different systems. A situation of a system includes the state, attribute, and state transition from

the state perspective, which changes with the safety behavior interactions.

3.1 Safety state and attributes modeling

(1) Safety State Definition.

Definition 2: State describes the safety conditions of a block at a particular moment, and defines how the block will respond to event occurrences [4]. A state can cover the normal operational modes, as well as the abnormal failure modes.

Each state is allocated with a flag value, which has a value of “1” for the normal state and a value of “0” for failure states.

Fig. 5 Inner structure of an IBD

In the function level, the abnormal state is equivalent to the functional failure mode, including function loss and malfunction. In the subsystem-level function, the failure mode is extended based on the upper-level functional failure mode, including premature and late acceptance of information, incorrect data reception, no data reception, and so on. The failure mode of a component is determined based on the type and characteristics of that component, which can be found in the FMEA table or related reliability documents.

(2) Safety attribute definition

An attribute defines a structural property of a class. The description is composed of visibility, name, type, and a multiplicity.

Definition 3: Attribute represents a structural feature of a block. The description is composed of name, type, visibility, and multiplicity [15]. The attribute could be assigned with values defining the object state.

The attributes can be categorized as local and global attributes based on operational scenarios. The global

Fig. 6 Example of behavior model

Table 1 Model relationship specifications

Target Source	System	Function	Component	Behavior
System	Association			Dependency
Function	Composition to whole	Association	Allocated	Dependency
Component	Composition to whole		Association	Dependency
Behavior				Association

attributes can be shared between all simulation objects and can also be dynamically updated during the simulation. In contrast, local attributes are exclusively accessible by an object itself, and they can be considered as internal system parameters. The local attributes can be called during the simulation, and they can respond to the changes in the other entities' behaviors and global attributes.

Furthermore, the attributes can also be divided into parameter attributes, state attributes, and performance attributes according to the applications. The parameter attributes denote functional parameters and data flow. The state attributes reflect system state conditions, where value "1" indicates normal state, value "0" indicates failure state, and Sect. (0, 1) denotes the degraded state. Besides, the performance attributes describe the system operational performances, such as liquid pressure and liquid temperature in mechanical systems and CPU occupancy and memory usage of electronic systems.

(3) State transition specification

Definition 4 State transition defines when and how state changes from one to another, considering the integrated influ-

ence of attributes, current states, and external events. Each state transition is characterized by three optional categories of information: a trigger, a guard, and an effect [4]. Those categories of information can be presented in a single string as follows: <trigger>[<guard>] / <effect>.

The trigger denotes the conditions of certain properties of a parameter or state attributes. The state transition will be performed when the trigger is satisfied. The guard provides the possibility to suppress state transitions in certain situations. The effect focuses on behaviors to be executed during the state transition. As for the system safety, a trigger is regarded as a response interface to the occurrence of a particular abnormal external event, a guard determines the transition feasibility based on safety attributes, and an effect conducts some measures to mitigate the failure effect, and transmits the internal abnormal state to other objects to establish dynamic relationships between current states of different objects. Particularly, the safety state transmissions between the objects are realized by the following two safety behavior expressing mechanisms in Sect. 3.2.

Fig. 7 Model relationship framework

3.2 Dynamic expression of safety behavior

The event transmission mechanism and global attribute updating mechanism are designed to realize dynamic expressions of safety behaviors in various situations, which are then embedded into the effect information of state transitions within the safety state model layout.

(1) Event transmission mechanism

Definition 5: Event represents an abstract expression of the dynamic behavior, that defines a type of occurrence that can trigger a behavior within an operational system by interacting with the trigger, guard, and effect information in the state transition or the operations in the state block [4].

An event could occur several times during operation, capable of triggering a new execution of a behavior at each time if the conditions are satisfied.

To establish the association of various events with all the objects in the modeling environment, the event transmission mechanism is designed based on the effect information of the state transition. In the event transmission mechanism, the BROADCAST_EVENT & SEND_EVENT structure is adopted to transmit the effect event of a certain state transition to other simulation objects in either non-directional or directional form, which is expressed as:

(1)%BROADCAST_EVENT("event")%

(2)%SEND_EVENT("event",CONTEXT_REF(target))%

For example, the effect caused by the state transition from “Function_normal” state to “Malfunction” state is abstracted as the event “FUNC1_FAIL”, which is then spread to the other objects in the simulation environment by the event broadcast gramma, as shown in Fig. 8.

Particularly, an event propagates to the object system in the form of an event queue and then dispatches to the current activated states during the next simulation step, which is further functionalized to the entry/do/exit behavior of the state block.

(2) Global safety attributes updating mechanism

The global attribute updating is another effective method for the description of the interaction between states. The global attributes can be assigned directly by a specific event or updated indirectly as a reaction to the safety behavior inside the block. The value of a global attribute influences various local attributes and safety states of different objects from a global perspective.

To establish the principle of the global attribute and state updating, a time sampling state machine is especially designed, which is capable of updating the global attributes and synchronizing states according to a certain sampling interval during the simulation. Then the relationships between global attributes and local attributes are specified by equations inside the state blocks. On this basis,

Fig. 8 Example of the event transmission mechanism in the state machine

Fig. 9 Aircraft integrated surveillance function chain at the aircraft level

a certain global attribute could couple with the local attributes inside the block and the other global attributes in the simulation environment to refine the current state further.

4 Simulation verification

The aircraft integrated surveillance system (ISS) is used to verify the feasibility of the proposed method. The ISS represents a functional combination of traffic collision avoidance system, terrain awareness and warning system, and weather radar, which is capable of monitoring the traffic, terrain, and weather condition around the aircraft and provide the flight crew with collision avoidance alert and motivation advice. In particular, the traffic surveillance function was the research objective, and the required modules of the internal and exter-

nal systems were modeled. Then typical scenarios were set considering the inherent failure mode of the component, as well as the joint failure of different components, to validate the proposed method by observing the failure effect.

4.1 System model establishment

By being centralized on environmental surveillance, the aircraft function chain, system interaction relationship, system decomposition, hierarchical physical structure, and state machines from the functional level to the component level are shown in Figs. 9, 10.

Multiple aircraft level functions, including providing environmental surveillance, electrical power, and information display, were coupled to accomplish the flight task of environmental surveillance, as shown in Fig. 9. Then aircraft level

Fig. 10 System interactions related to the integrated surveillance function

Fig. 11 Integrated surveillance system decomposition

systems shown in Fig. 11 interacted with each other physically to accomplish the above-mentioned aircraft functions. Once the aircraft level relationships had been constructed, the protagonist was transferred to the integrated surveillance system, following the layer-by-layer refinement of the functional and physical decomposition, which complete the static structure modeling of the normal operation, as shown in Figs. 12, 13. Figure 14 depicts that the traffic collision avoidance system module (TCAS) is embedded in the integrated

surveillance system processing unit (ISSPU). Furthermore, the behaviors ranging from the aircraft functional level to the component level were portrayed by the state machine diagrams. The state machine of the traffic surveillance process describing the state transformation attributing to internal fail or external effect, such as electrical power, is shown in Fig. 15.

Enterprise Architect (EA) software [6] was used to construct the system model. This software supports the dynamic

Fig. 12 Integrated surveillance system's physical structure (subsystem level)

Fig. 13 Integrated surveillance system's physical structure (component level)

simulation of state machines, providing a powerful means of rapid generation, simulation, and visualization of complex state models, as depicted in Fig. 10. Moreover, the state transition processes can be recorded by timing diagrams which are generated automatically during the simulation.

4.2 Simulation results analysis

4.2.1 Inherent failure situation: traffic antenna fail

The ISS system obtains the surrounding aircraft information by a traffic antenna with four-unit bodies. The radio

Fig. 14 Traffic surveillance process state machine

Fig. 15 Enterprise Architect simulation environment

Fig. 16 Radiation diagram of a four-unit traffic bodies antenna

frequency (RF) signals transmitted by the four-unit bodies are all 90-degree lobes, constituting together a 360-degree omnidirectional detection range around the aircraft, as shown in Fig. 16.

In specific, the fault of the antenna feeder was simulated. It was assumed that the resistance of the TCAS antenna feeder R increased by R_T due to the failure of the connecting device (such as the loosening of the bolt), and the resistance of the feeder changed to R_1 , which is given by:

$$R_1 = R + R_T \quad (1)$$

The RF current on the feeder line was denoted as I , and the RF current on the feeder line with the pin loose was denoted as I_1 . As the RF voltage U output by the computer was constant, the RF current on the feeder changed as follows:

$$I = U/R \quad (2)$$

$$I_1 = U/R_1 = U/(R + R_T) \quad (3)$$

Thus, when R_T was much larger than R , then I_1 was smaller than I , and the power P could be obtained by the power as $P = UI$. The radio frequency power $P_1 = UI_1$ acting on the TCAS radio frequency line was far less than the normal-condition power $P = UI$. The distance of the RF signal emitted by feeder J_2 was initially γ and changed to γ_1

after the pins loosening of the pins, and they were, respectively, expressed as:

$$\gamma = \chi \times P \times \sin \alpha \quad (4)$$

$$\gamma_1 = \chi \times P_1 \times \sin \alpha \quad (5)$$

where χ denoted the spatial attenuation coefficient. The distance γ of the RF information emitted by feeder J_2 reduced sharply due to the reduction in the transmit power on the feeder.

After the local feeder failure of the 1/4-lobe of the antenna, the increase in the feeder resistance passed to the TCAS module within ISSPU through the synchronous update of the global attributes. As the TCAS module possessed the built-in fault detection mechanism, the gradients of P less than a given threshold appeared as a certain failure mode, and the state transition event was broadcasted to the other models. The state machines of the traffic antenna and TCAS module of the ISSPU are shown in Fig. 17a, b, respectively. The timing diagrams of the affected modules and functions are illustrated by the rest sub-diagrams in Fig. 17.

Due to the injected failure of the traffic antenna at 9 s shown in Fig. 17c, the state of the receiving module of the TCAS module in the ISSPU degraded at 14 s because of the influence of the antenna signal, as shown in Fig. 17d. Therefore, the state of *Func5_1_send_receive_signal* turned

Fig. 17 Simulation results of the traffic antenna failure

Fig. 18 Simulation results of the TCAS module failure and state monitor sensor failure

to a degraded state (the state value was coded as 3/4 according to the antenna configuration). Similarly, the state of function *Func5_2_Signal_processing_and_alert_calcu* was degraded subsequently (the state value was coded as 3/4). The entire ISSPU safety state degraded at 17 s after the TCAS module failure occurred, as shown in Fig. 17e. The top-level function remained operating, since the hazard effect was moderate due to the bode redundancy, which is demonstrated as degraded state in Fig. 17f. Besides, the state calculation results of the logic Eqs. (6) and (7) also indicated that the top-level traffic surveillance function could be completed as the task state value remained larger than zero.

$$\begin{aligned}\text{Func5.state} &= F(\text{Func5_1}, \text{Func5_2}, \text{Func5_3}) \\ &= \text{Func5_1.state} \times \text{Func5_2.state} \times \text{Func5_3.state} \\ &= 0.75 \times 0.75 \times 1 = 0.5625\end{aligned}\quad (6)$$

$$\begin{aligned}\text{Task.state} &= F\left(\sum_{i=1}^6 \text{Func_i}\right) = \prod_{i=1}^6 \text{Func_i.state} \\ &= 1 \times 1 \times 1 \times 1 \times 0.5625 \times 1 = 0.5625\end{aligned}\quad (7)$$

4.2.2 Joint failure situation: equipment reset mechanism fail

Equipment reset mechanism aims at resetting the abnormal equipment when the failure is detected, to recover the expected function of the equipment. The effectiveness of the mechanism lies in the capacity of fault detection and the reliability of the reset mechanism. This scenario simulated the joint failure of the TCAS module and state monitor sensor that performed the failure detecting function by monitoring ISSPU state.

During the simulation process, a certain threshold was set for the reliability of the state monitor sensor. Then in each sampling period, a random variable was generated and compared with the threshold. A latent failure of the monitor sensor occurred when the threshold was exceeded. As shown in Fig. 18b, sensor failure was randomly triggered at 20 s.

Besides, internal failure of the TCAS module of the ISSPU occurred at 9 s and 30 s, as shown in Fig. 18c, which induced the internal failure of the ISSPU.

In the first failure case, as the sensor managed to detect the failure, the equipment reset mechanism was conducted, which took 7 s due to the internal calculation cost. Then the states of the TCAS module and ISSPU were reset to normal at 16 s, as shown in Fig. 18c, d. However, in the second case, when the ISSPU failure occurred again, the loss of fault detection capacity led to the failure of the entire traffic surveillance function at 30 s, as depicted in Fig. 18e.

5 Conclusion

Aiming at a dynamic and systematical evaluation of failure effect on the safety under various situations, this paper proposes a state-based dynamic safety analysis method based on an executable state machine. The feasibility and correctness of the proposed method are verified by simulations with the aircraft integrated surveillance system via the dynamic failure simulation and failure effect analysis.

The main contributions can be summarized as follows:

(1) A multi-layer and multi-system state interaction framework is constructed realizing the interaction between structural models and behavior models from the aircraft level to the component level.

(2) The event transmission mechanism and the global attribute updating mechanism capable of expressing the safety behavior dynamically and globally in various situations are developed, thus achieving evaluation of failure effect globally.

The proposed method extends the safety analysis from the perspective of the situation and realizes the global perception of the safety state under specific failure mode, promoting the dynamics and integrity of the safety analysis process. In future research, the inherent safety handle mechanism modeling will be complemented to the modeling framework, thus elevate precision of the failure effect analysis progressively.

Acknowledgements This paper is sponsored by National Program on Key Basic Research Project (2014CB744903), National Natural Science Foundation of China (61673270), Natural Science Foundation of Shanghai (20ZR1427800), New Young Teachers Launch Program of Shanghai Jiaotong University (20X100040036), Shanghai Pujiang Program (16PJD028), Shanghai Industrial Strengthening Project (GYQJ-2017-5-08), Shanghai Science and Technology Committee Research Project (17DZ1204304) and Shanghai Engineering Research Center of Civil Aircraft Flight Testing.

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

References

1. Batteux M, Prosvirnova T, Rauzy A (2017) AltaRica 3.0 assertions: the whys and wherefores. *J Risk Reliab* 231(6): 691–700.

2. Biggs G, Sakamoto T, Kotoku T (2014) A profile and tool for modelling safety information with design information in SysML. *Softw Syst Model* 15:147–178
3. Chen X, Jiao J (2017) A fault propagation modeling method based on a finite state machine. *Proc Annu Reliab Maintainability Symp* 0: 1–7.
4. Delligatti L (2013) State Machine Diagrams. In: *SysML Distilled: a brief guide to the systems modeling language*. Addison-Wesley Professional, pp 155–173.
5. Ericson CA (2005) Hazard analysis techniques for system safety. N.J.Wiley-Interscience, Hoboken
6. Ferreira TB, Stempluc SM, Lisboa-Filho J (2014) Geographical data modeling with the UML GeoProfile and MDA transformations on the Enterprise Architect Tool. In: *Proc 9th Iberian Conf Inf Syst Technol*, pp 1–6.
7. Hönig P, Lunde R, Holzapfel F (2017) Model based safety analysis with smartIfflow. *Information* 8(1):7–22
8. Lisagor O, Kelly T, and Niu R (2011) Model-based safety assessment: review of the discipline and its challenges. *Proc Int Conf Reliab Maintainab Saf*: 625–632.
9. Lunde K (2003) Ensuring system safety is more efficient. *Aircr Eng Aerosp Technol* 75:477–484
10. Müller C, Hönig P, Lunde R (2018) Evaluation of smartIfflow based on the wheel brake system from ARP4761. *Int Fed Autom Control* 51(24):1255–1262
11. Papadopoulos Y, Walker M, Parker D et al (2011) Engineering failure analysis and design optimisation with HiP-HOPS. *Eng Fail Anal* 18:590–608
12. Rauzy A, Blériot-Fabre C (2014) Model-based safety assessment: rational and trends. *Proc Eur-Asia Congr Mechatronics*: 1–10.
13. Roth M, Gehrlicher S, Lindemann U (2015) Safety of individual products-perspectives in the context of current practices and challenges. *Proc Int Conf Eng Design* 3:113–122
14. Sharvia S, Papadopoulos Y (2015) Integrating model checking with HiP-HOPS in model-based safety analysis. *Reliab Eng Syst Saf* 135:64–80
15. Weilkens T (2007) *Systems engineering with SysML/UML*. Morgan Kaufmann OMG Press, Burlington
16. Würtenberger J, Klobardanz H, Lotz J, Ahsen A (2014) Application of the FMEA during the product development process—dependencies between level of information and quality of result. *Proc Int Des Conf*: 417–426.