

Using Model-Based Security Engineering in the Development of Complex Aircraft Cabin Systems

Hartmut Hintze and Ralf God
Hamburg University of Technology

ABSTRACT

The increasing functionality associated with the rising complexity of aircraft cabin systems which are used by cabin crew, passengers, maintenance staff and other stakeholders, requires a reconsideration of the methods for the development of aircraft cabin systems. This paper deals with a model-based security engineering approach based on the so called Three-V-Model as an appropriate process model, which represents the governing system engineering process (SEP) associated with the safety engineering process (SafEP) and the security engineering process (SecEP). All three processes are pursued concurrently and are interacting reciprocally by working within the same system model on each development level. We describe in detail the involved model-based security engineering activities of the SecEP and the integration of the CORAS risk analysis method in a consistent System Modeling Language (SysML) approach. Finally we demonstrate how the interactions between the SEP and the SecEP with the CORAS risk analysis method are realized within a single SysML model.

CITATION: Hintze, H. and God, R., "Using Model-Based Security Engineering in the Development of Complex Aircraft Cabin Systems," *SAE Int. J. Aerosp.* 8(1):2015, doi:10.4271/2015-01-2445.

INTRODUCTION

The cabin management system plays a central role in all operational tasks within the aircraft cabin. Primary functions are communication, indication, control, monitoring and configuration of other cabin systems. For this reason it represents a complex system characterized by a large amount of interactions between a plurality of aircraft cabin systems. The constantly growing number of functions associated with the equally increasing complexity of modern cabin systems requires a new consideration of adequate methods for their development.

Nowadays the development is covered by requirements-based engineering (RBE), which is characterized by a requirements-driven proceeding and computer-assisted tools for requirements management [1-2]. This RBE approach, based purely on textual documentation, is rapidly reaching its limits in balancing quality and complexity for the next generation of cabin management systems. In order to control the complexity within the system engineering process and to provide early detection of possible design flaws, the method of so-called model-based requirements engineering (MBRE) promises remedy [3]. The central thought of MBRE is to produce abstract system models at an early stage within the development process which represent the different aspects of a system comprehensively. The objective is that the models produced by MBRE can represent the actual and complete system specification. Text documents are only used in addition, in order to specify characteristics which cannot be illustrated in the format of the model.

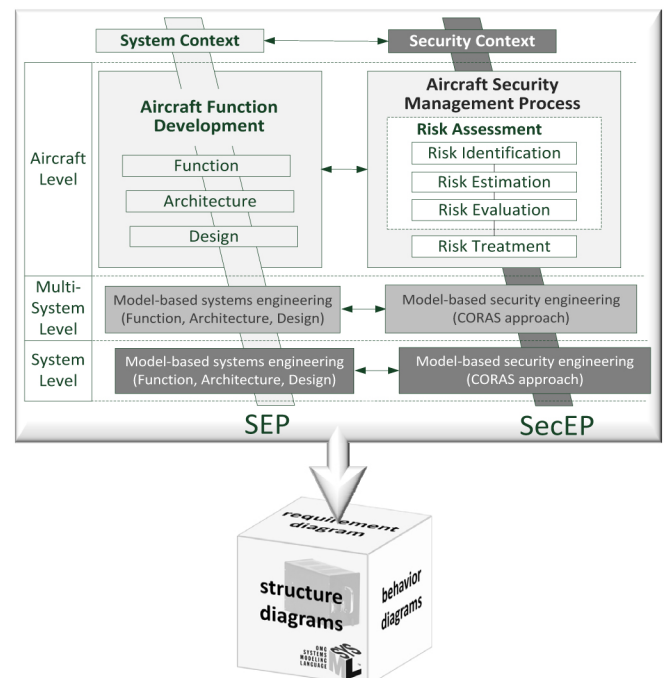


Figure 1. Model-based execution of particularized SEP and SecEP activities within the design phase according to the Three-V-Model [5] and their documentation in a single and unified SysML system model.

Based on the Three-V-Model [4-5], which represents the governing system engineering process (SEP) associated with the safety engineering process (SafEP) and the security engineering process (SecEP) this article describes a model-based security engineering

approach to be used for the development of complex aircraft cabin systems. As shown in [Figure 1](#) we show in detail how the particularized SecEP activities of the Three-V-Model [5] can be executed in a model-based context within the design phase. Further we explain the integration of the SEP and SecEP into a single and unified SysML system model which is symbolized as a cube shown in [Figure 1](#). The sides of the cube represent different views on the system, e.g. on requirements, structure or behavior, where a block diagram for instance provides information on system structure or a use case diagram on system behavior.

THE AIRCRAFT CABIN AND CABIN SYSTEMS

Within the past 20th century engineering work in commercial aviation was focused on improving the performance parameters to achieve higher payload, extended range, higher speed and more comfort for the passengers. Important milestones were set by major achievements in aircraft design, engine technology and by the use of pressurized cabins for higher cruise altitudes.

Today, in the 21st century, the focus has moved towards the aircraft cabin. Within air travel the cabin is the central element for operations and has to showcase airline reputation and brand personality. The primary transport service of an airline is accompanied by a fierce competition for customers who force the airline to deal intensively with passenger's future travel requirements. An increasing range of cabin functions, maintenance tasks and passenger services as well as a related growing number of processes are now culminating in a challenging situation where the existing architecture needs to be reconsidered. Even the system development method should be reviewed in terms of controlling rising complexity of cabin systems to ensure the retention of next generation cabin management system quality within the scopes of its function, its safety and its security based on new architecture options.

Today's Development Process of Cabin Systems

The development of current cabin systems follows three instances of the V-Model [4-5] as shown in [Figure 2](#). The first V represents the fundamental system engineering process (SEP) for the development of aircraft functions and is documented in literature [6]. The branches of the V are symbolizing the three major phases of system design, implementation and system integration.

The second V accompanies and supports the basic SEP and represents the safety engineering process (SafEP). This process is well known from literature [7] and focuses on system safety requirements and airworthiness design, i.e. on reliability and fault tolerance topics.

To ensure that the security requirements for novel aircraft cabin system architectures are visualized and encountered more comprehensively at an early stage during the development process an accompanying and supporting third V was introduced. This V represents the security engineering process (SecEP) [4-5] which is becoming increasingly important. A generic specification for a SecEP is given in [8]. To cope with all particular guidelines of the aircraft development process we further on refer to this Three-V-Model as a

comprehensive process model. In this model the SEP for the development of system functions, the SafEP covering system reliability and the SecEP to protect the system from attack and misuse are pursued in a concurrent way.

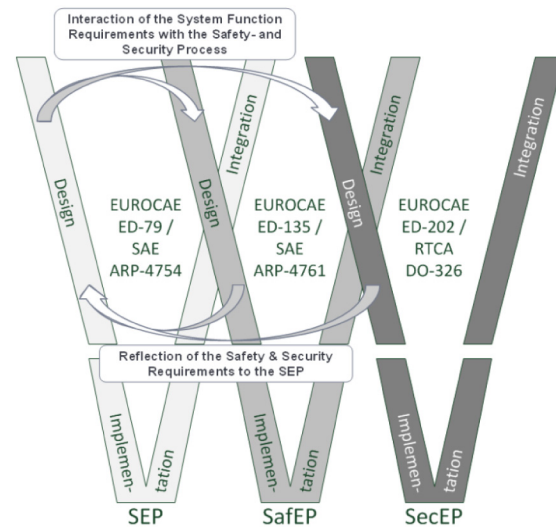


Figure 2. The Three-V-Model derived from EUROCAE, SAE and RTCA guidelines [6, 7, 8] for the systems engineering process (SEP), the safety engineering process (SafEP) and the security engineering process (SecEP).

Today the development process is implemented via the requirements-based engineering (RBE). This text-based methodical procedure of RBE is increasingly reaching its limits in guaranteeing the quality of next generation complex cabin management systems. Textually specified characteristics and functions provide a barrier to obtaining a detailed overview and an accordingly deep consideration of system functions, safety and security. For a development engineer it is challenging to survey the sheer number of elements to be considered within today's aircraft design and difficult to understand the reciprocal effects when faced with a purely textual documentation. So-called model-based requirements engineering (MBRE) can be used in the development of hardware and software-based complex systems to not only facilitate and improve the process but to also provide early detection of possible design flaws [3]. MBRE links the text-based requirements management approach with the model-based system engineering (MBSE) procedure [9], which is described within the following section.

Model-Based-Requirements-Engineering (MBRE)

The central concept of MBRE is to follow the principles of the RBE with its requirement management and to produce an abstract system model at an early stage of the development process. This model comprehensively represents the different aspects of a system such as functions, safety and security [6, 7, 8]. An elaborated system model following the MBRE concept can basically fully replace a classical system specification formulated in natural speech.

Formal modelling languages, such as e.g. the system modelling language (SysML) [10] or the unified modeling language (UML) [11], permit a standardized system description [9] and create a so-called formal specification which permits a formal verification of

the system requirements. On the one hand the use and approaches of MBRE are well described in literature [3], however these focus primarily on the SEP and system functionality. On the other hand there exist initial approaches for the modelling of system security [12], but these mostly omit the implementation of a security engineering process. This paper therefore demonstrates how a security engineering process (SecEP) can be arranged and implemented by means of MBRE.

The current SecEP in aviation comprises the risk management process following the ISO/IEC 31000 standard [13]. For this reason the selected model-based risk management approach should also follow this standard to support a successful implementation of a model-based SecEP. Moreover the selected approach should be compatible to common modelling languages, such as SysML or UML to be in line with already existing SEP approaches, which are exemplarily described in [5]. Considering these two major prerequisites the CORAS approach [14] for model-driven risk analysis was selected as a suitable way forward.

The CORAS Approach

The CORAS approach [14] comprises a method to perform a model-driven risk analysis and has been developed by empirical investigations and a series of industrial field studies. The development was financed by the Norwegian Research Council together with the EU and is based on the ISO/IEC 31000 Risk management - Principles and guidelines (cf. Figure 3) [13].

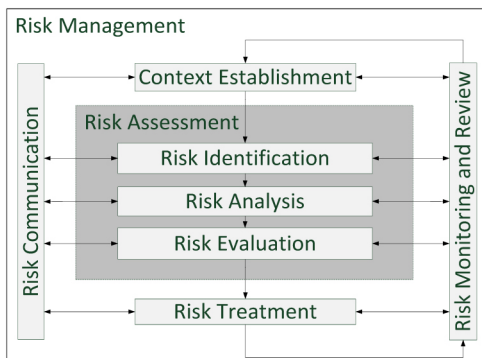


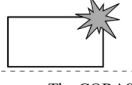
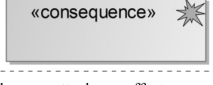
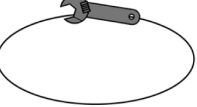
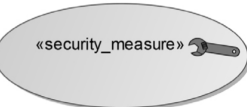
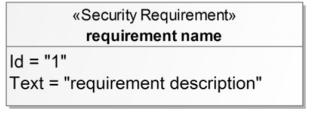
Figure 3. Security Risk Management Process defined in ISO/IEC 31000 [13].

CORAS uses specific diagrams inspired by the UML to document the results of the risk assessment and to present the overall risk management conclusions. In line with the ISO/IEC 31000 standard the CORAS approach defines five analyzing steps which are supported by different model-based diagram types as follows:

1. *Context Establishment* is to identify stakeholders as well as assets, and to decide which parts of the system need to be taken into account. According to the CORAS method the asset identification is described via an asset diagram.
2. *Risk Identification* is to identify potential threat scenarios that may constitute risks. According to the CORAS method the threat scenario identification is described via a threat diagram.
3. *Risk Analysis* is to estimate likelihood and impact for the risks that have been identified. Likelihood and impact are described via a threat diagram.

4. *Risk Evaluation* is to prioritize risks according to their severity in order to identify those which will be subject to treatment. The outcome is described by a risk diagram.
5. *Risk Treatment* is to propose and specify appropriate treatments. At proposal level the specification is done via a risk treatment diagram. The final accepted risk treatments will be resumed through a treatment overview diagram.

Table 1. Conversion of CORAS symbols into SysML compatible elements.

CORAS symbol	converted element (SysML)
	 «origin_human_threat_deliberate»
Origin: The CORAS symbol describes the source of a system attack. The adequate SysML element is the SysML "actor", which was graphically adapted to the CORAS style.	
	 «direct_asset»
Asset: The CORAS symbol describes the value of a system that is affected by the consequence of an attack. The adequate SysML element is the SysML "actor", which was graphically adapted to the CORAS style.	
	 «cause»
Cause: The CORAS symbol describes the way how an existing vulnerability is used to attack the system based on the origin. The adequate SysML element is the SysML "use case" and addresses the misuse case of a system.	
	 «consequence»
Consequence: The CORAS symbol describes how an attack can affect an asset. The adequate SysML element is the SysML "block", which was graphically adapted to the CORAS style.	
	 «vulnerability»
Vulnerability: The CORAS symbol describes the security gap which enables the attack of a system. The adequate SysML element is the SysML "actor", which was graphically adapted to the CORAS style.	
	 «security_measure»
Security Measure: The CORAS symbol describes the countermeasure which is implemented within a system to prevent a successful attack. The adequate SysML element is the SysML "use case" and addresses the security use case of a system.	
	 «risk» +SecR = risk level:
Risk: The CORAS symbol describes the risk level for a successful attack. The adequate SysML element is the SysML "block", which was graphically adapted to the CORAS style.	
no symbol defined	 «Security Requirement» requirement name Id = "1" Text = "requirement description"
Security Requirement: CORAS does not provide a security requirement symbol. A security requirement specifies the need of a security measure which is not implemented yet and is given by the result of the risk analysis within the SecEP. The adequate SysML element is the SysML "requirement", which was implemented as a new SysML stereotype.	