AST 2015
Workshop on Aircraft System Technologies

MODELING TOP-LEVEL REQUIREMENTS FOR A TANGIBLE USER INTERFACE IN THE AIRCRAFT CABIN

Thorsten Kiehl, Ralf God

Institute of Aircraft Cabin Systems, Hamburg University of Technology
Nesspriel 5, 21129 Hamburg, Germany

thorsten.kiehl@tuhh.de

Abstract
Systems development in the aircraft industry is characterized by considerably long timelines, a high complexity and a predominantly text-based and document-centered approach. Misinterpretation of requirements and design flaws in early development phases can cause remarkable delays and adverse effects on cost in later phases. A promising approach to deal with complexity and to avoid those misinterpretation and mistakes is the model-based requirements engineering (MBRE). However the potential of this methodology is seldom used in the initial requirements phase of aircraft systems. User interfaces of in-flight entertainment and communication (IFEC) systems are advancing rapidly and have become a decisive factor for airlines to differentiate from their competitors. Thus a comprehensive and more capable methodology for the elicitation, analysis, specification and verification of top-level requirements is needed. This paper presents a recipe for an efficient model-based approach. The usability of this methodology is demonstrated by modeling top-level stakeholder and system requirements for an IFEC system with a tangible user interface in the aircraft cabin.

1 INTRODUCTION

From a user perspective there should be no significant difference when using transmitting portable electronic devices (T-PEDs) at the airport and in the aircraft cabin. However, technology gaps between these two worlds do certainly exist. The development of aircraft cabin systems with novel user interfaces is complex and certification issues lead to long development times, which often exceed a period of ten years. It is challenging to deal with the multitude of requirements which have to be analyzed, specified and validated from an aircraft systems perspective and the user perspective. Information and communication systems at the airport, in contrast, are very strongly

oriented towards the portable electronic equipment of passengers. Consumer electronics with short development cycles and new products every year are rapidly changing the user behavior. Keeping pace with the expectations of passengers is ambitious in the cabin facing longtime development cycles of aircraft systems. A known example is the moderate functionality of an aircraft built-in in-flight entertainment and communication system (IFEC) in contrast to the manifold applications and personalized functionalities of a passenger's smartphone for communication, entertainment and even more for the convenient control of other systems and peripherals.

For enhancing and supporting the challenging systems engineering process, industries already rely on modeling and simulation of systems for quite some time. This modeling in combination with a structured procedure for requirements elicitation, analysis, specification, verification and validation is known as model-based requirements engineering (MBRE) methodology, which enables a directed and comprehensive system development lifecycle for complex systems. In aviation industry requirements engineering is still mostly done purely text-based and document-centered. Using the potential of modeling and simulation is often limited to the later system design and implementation phases. In this way noteworthy optimizations and savings are already attainable [1], but the complete potential of MBRE can only be exploited by a comprehensive application at an early requirements elicitation stage.

Therefore it is a must to perform the top-level requirements specification model-centered. In contrast to low-level requirements, those top-level requirements represent the stakeholder needs and define the general system behavior. When top-level requirements are specified a detailed system and component structure is not available or in a focus. Modeling requires the usage of a capable and flexible modeling language like the semi-formal Systems Modeling Language (SysML) as well as supporting modeling software (typically called "tool"). Notably an efficient method for the structured process of MBRE is important.

The development of a tangible user interface in the aircraft cabin based on Near Field Communication (NFC) was selected to be one of the first projects which should benefit from the application of a structured MBRE-approach in the initial and fundamental requirements phase. More and more smartphones and T-PEDs are equipped with an NFC interface, and even Apple, Inc. as the last major manufacturer of high-value smartphones changed their strategy and implemented an NFC interface in their latest product [2]. A significant effort of the banking industry in updating their terminals with NFC capabilities will now continuously change payment procedures on ground. For airlines this novel "touch and pay" procedure should be of major interest, because an easy and intuitive way of cashless micro-payment supports their need to generate ancillary revenue during flight. Currently there is only a limited number of NFC trials in aviation industry and most of them focusing on applications on ground. Recently the authorities in Europe and the United States loosened their regulations for PED-usage aboard [3], which raises the question of applications and usage in the aircraft cabin. Integrating NFC interfaces in the aircraft cabin involves numerous stakeholders with a remarkable set of top-level requirements. An MBRE-approach with a

model-centered requirements phase is able to support the sophisticated and detailed requirements specification and management with always having the look at the big picture. Based on this approach and a tailored framework presented in [4], this paper will follow up and contribute to an integrated and model-centered work procedure.

2 THE METHODOLOGY

In systems engineering a methodology is defined as a collection of related processes, methods and tools [5]. Together these elements form a kind of recipe to solve a given engineering task. A common process framework and its use for systems engineering is defined by the ISO/IEC-15288:2008 standard "Systems and software engineering – System life cycle processes" [6]. As this standard is commonly used and well accepted, the methodology in this paper will be based on the processes defined there. There are two underlying technical processes for specifying top-level requirements: The *Stakeholder Requirements Definition Process* and the *Requirement Analysis Process*. A process defines "what" is to be done, without specifying "how" a task is performed, whereas a method consists of techniques for performing a task and defines the "how" of each task [5].

Such kind of method is described by Holt, Perry and Brownsword [7], also known as *Approach to Context-based Requirements Engineering* or *ACRE* ontology. As the name states, the key concept of the *ACRE* ontology is strictly focusing on the context. Requirements are only given meaning, when there is at least one use case which describes their context. Use cases in turn describe needs and expectations of the stakeholders, which might be manifold and diverse and even opposing requirements can initially occur. To proof the significance of a use case, each use case shall be validated by a scenario. A requirement is always elicited from somewhere, so it should be traceable to its so called source element, which can be every type of information, for example a book or an article, but also a meeting protocol or a list of standards. A more elaborated coaction of all elements within the *ACRE* ontology [7] is shown in Figure 1.

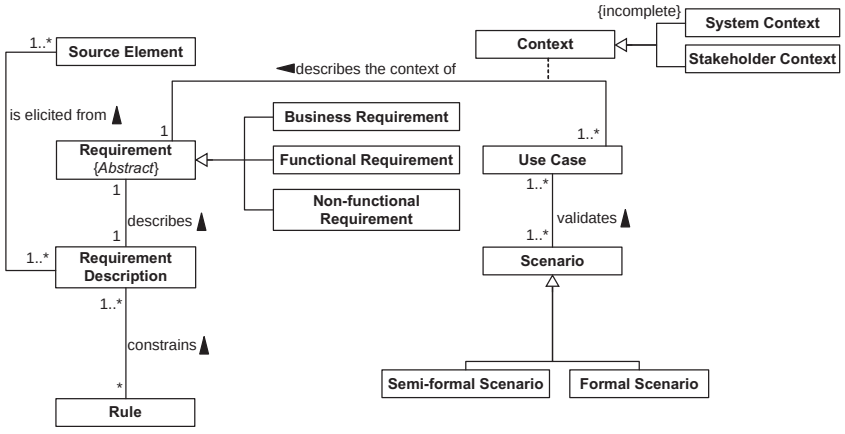

Figure 1 – ACRE-Ontology by Holt, Perry and Brownsword [7]

Before starting to model a major project in SysML, there is the need for tailoring to the intended modeling process, because SysML itself does not provide detailed information for the correct application of language elements. Activities which lead to a tailored framework were already described in [4]. Having a tailored framework available, the model environment needs to be set up. This sequence of required activities is depicted in Figure 2. With the model structure on-hand, the processes of the ISO/IEC-15288:2008 standard can be executed, starting with the elicitation of stakeholder requirements. By subsequently following the *Requirements Analysis Process*, system requirements are derived. Thus the top-level requirements are defined and set the basis for further systems engineering phases like designing the architecture and finally implementing the system.

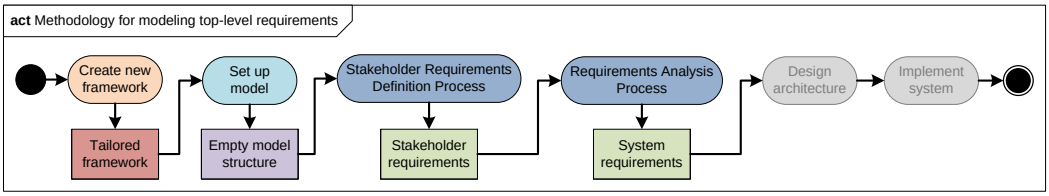

Figure 2 – Methodology for modeling top-level requirements

3 STRUCTURE THE MODEL

A major advantage of MBRE is the comprehensive information storage in a dedicated model repository. This enables storing all elements' relations and an easy jumping between the model elements. To be able to focus on individual details and not being distracted by the whole complex thing, as well as being able to orient oneself in the model, a well-structured repository is indispensable. A leading example for a wise model partitioning is published by the SE^2-Challenge Team with their *Active Phasing Experiment (APE)* model [8]. Their model is structured by using SysML *packages* and follows a recursive package structure along the system breakdown. As this structure fully complies with the requirements of the *ACRE* ontology, most parts of the *APE* structure were adopted for this paper's methodology.

The overall breakdown of the model, i.e. the model structure for the project example "tangible user interface in the aircraft cabin" is depicted in Figure 3. On project level the model package is interlinked with the model library and the profiles (cf. Figure 3 left). The model library contains all information that can also be used in other models. These are general requirements like standards and laws as well as a set of physical units and dimensions to define interfaces. The profiles contain stereotypes and model customizations defined during the tailoring process. There are several packages on each system level which correspond to the elements of the *ACRE* ontology (cf. Figure 3 middle and right): "Requirements", "SourceElements", "Context", "Behavior" (for use cases and scenarios), "Traceability", "Verification" and "Structure", where "Structure" is the container for the next detailed system level having a recursive

set of packages. Using customized stereotypes in each diagram helps to quickly identifying its function. By adding hyperlinks and additional buttons to the elements and diagrams an easy navigation within the model structure is possible.

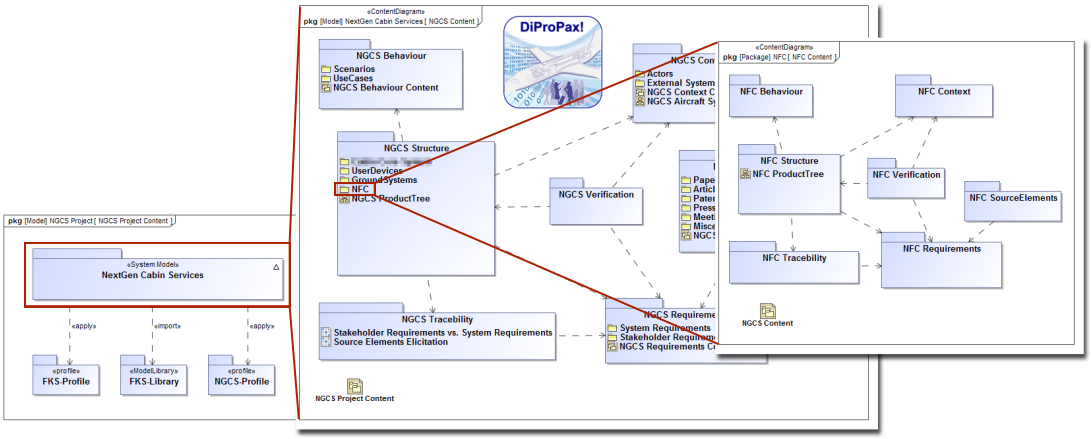

Figure 3 – Project level packages (left) and package structure on system level (middle) and recursive subsystem level (right)

4 STAKEHOLDER REQUIREMENTS DEFINITION PROCESS

The ISO/IEC-15288:2008 standard states the purpose of the *Stakeholder Requirements Definition Process* "to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment" [6]. This process can be divided into three phases: *stakeholder requirements elicitation*, *stakeholder requirements definition* and *stakeholder requirements analysis and maintenance*. The three phases and the related modeling of the comprised steps are depicted in Figure 4. Performing these phases in a subsequent order is feasible, but an iterative or recursive requirements engineering process is typically favored to include additional knowledge which may be gained during the process.

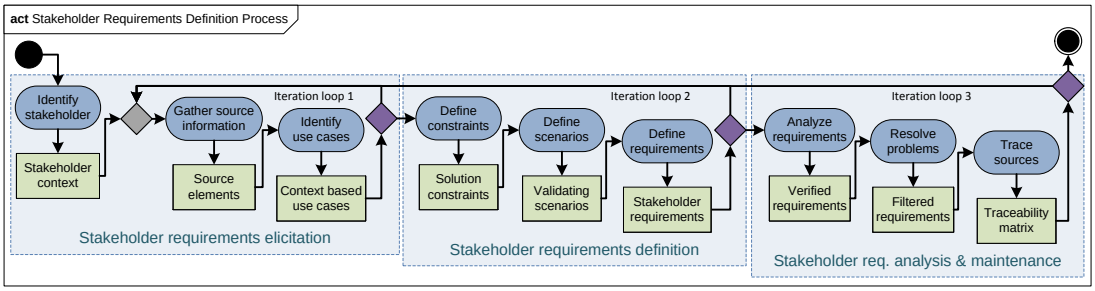

Figure 4 – Process for the model-based Stakeholder Requirements Definition

More in detail the first step in the elicitation phase is the identification of all involved stakeholders to define the stakeholder context of a planned system. This should be carried out very carefully because adding or removing a stakeholder with their needs

in a later project phase may influence the systems design dramatically. Stakeholders should be classified into different groups to highlight their origin and their interest in the system [7]. Using customized stereotypes (e.g. “User” and “Operator”) enables an easy identification of those groups [4]. In the model all SysML *actors* representing stakeholders are stored in the “Context” package [8]. When defining the stakeholder context for a tangible user interface in the aircraft cabin there are some obvious “User” like the passenger and the flight attendants and “Operator” like the airline, the airport and the payment provider. Stakeholders such as the aircraft manufacturer and equipment suppliers can be assigned to the “Supplier” group. The two stakeholder groups “Law” and “Standard” lead to requirements which regulate, restrict, constrain or standardize the system [7]. Regulations in the aircraft industry are mainly published by the authorities FAA (Federal Aviation Administration) and EASA (European Aviation Safety Agency). Other relevant standard setting bodies that might have an influence on the projected system are for example the Radio Technical Commission for Aeronautics (RTCA), the International Organization for Standardization (ISO) and the NFC-Forum [9]. The stakeholder context for the tangible user interface in the aircraft cabin with all interest groups is given in Figure 5.

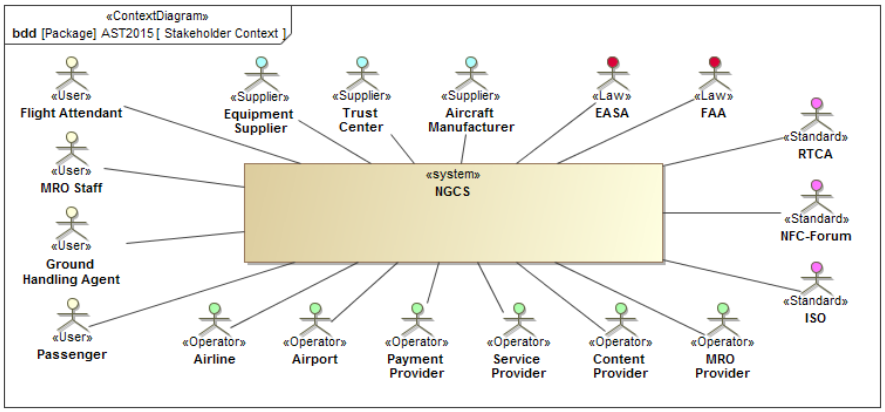

Figure 5 – Stakeholder context for a tangible user interface in the aircraft cabin

For the further two steps it is inevitable to gather as much information as possible to enable the understanding of the needs and expectations of all involved stakeholders. Interviewing them is a convenient method to obtain desired information, but that may not always be possible [7, 10]. Thus additional sources like articles, books, patents and workshop results should be included and utilized. Referencing such type of information in the model as source elements in the corresponding model package facilitates a later tracing and prevents from loss of information [7]. SysML does not directly have a dedicated language element to model source elements, but customizing a SysML *block* by stereotyping and adding additional tags (date, author, reference etc.) facilitates the capturing of all relevant information as source elements.

Based on those source elements it is now possible to identify use cases in the defined stakeholder context. They support the illustration of the ideas of each stakeholder and highlight the diverse interests [7]. For the tangible user interface it is a good way

forward to identify use cases of the passenger. Some of them are presented in Figure 6 left: a passenger for example wants to get personalized information about his connecting flight and/or wants to buy beverages or entertainment content by using his T-PED [11, 12]. In case of purchasing beverages this use case requires a direct interaction with the stakeholder "flight attendant" as the cabin crew has to deliver the beverages to the seat. The corresponding use case of the airline is generation of ancillary revenue. Bundling use cases to more generic ones helps to maintain an overview, which is supported by using customized stereotypes. In an iterative approach new source elements can be added to the model, e.g. if additional information comes up during the use case definition process (cf. Figure 4, iteration loop 1).

In the first step of the *stakeholder requirements definition* phase [6] constraints are identified. For a novel system there are often (management) constraints which interfere with other already existing systems or force the system to fit into a specific market gap. Such solution constraints are recorded during this step.

The elaboration of scenarios enables a validation of the previously identified use cases and generates a more in-depth understanding of the desired system behavior [7]. Together with the use cases these scenarios should be stored in the "Behavior" package of the model. The previous aggregation of use cases limits the amount of required scenarios. For the aggregated use case "consume local entertainment content" an example scenario can be described by using a SysML *activity diagram*, which is depicted in Figure 6 right: after opening the multimedia function on his PED the passenger can choose the media he wants to consume. If this media is free of charge, direct access is granted, otherwise a payment process has to be initiated. Using NFC for the payment process facilitates the purchase for the passenger at the point of sale.

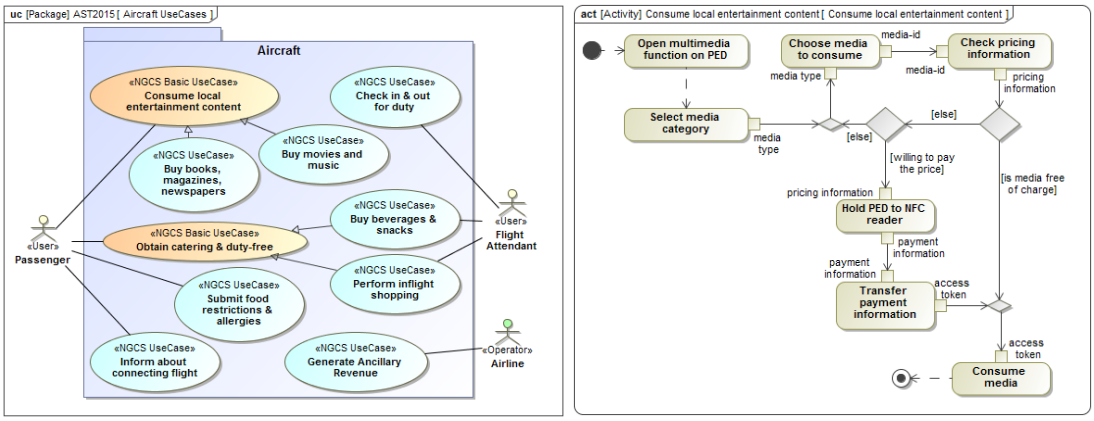

Figure 6 – Aircraft use cases with stakeholder context (left) and the validating scenario for consuming entertainment content (right)

After validating all use cases with scenarios, stakeholder requirements can be defined and put into the "Requirements" package on the system level of the model [8]. Tagging these requirements with a customized stereotype according to [4] helps to distinguish them from the later system requirements. Adding additional information to

the requirements, e.g. the source or the stakeholder, helps to easily identify the origin of a requirement [7]. Contradictory requirements arising from different stakeholder's contexts may occur during the process, however contraries do not have to be solved at this point and an iteration loop, gathering further source information, is feasible (cf. Figure 4, iteration loop 2).

In the subsequent *stakeholder requirements analysis and maintenance* phase the elicited requirements should be verified. The whole set of requirements has to be checked for conflicting, incomplete, ambiguous and inconsistent requirements [6]. In an intensive contact with all affected stakeholders those problems should be solved and potentially conflicting requirements should be prioritized. Requirements that are unrealistic to be realized or implemented should be discussed with the owning stakeholders. Finally all stakeholders have to agree with their set of requirements and - according to the *ACRE* ontology - all requirements should be traceable to their source elements [7]. Using a tracing matrix in the "Traceability" package enables a distinguished mapping of the relations.

5 REQUIREMENTS ANALYSIS PROCESS

The *Requirements Analysis Process* is the second process definition in the ISO/IEC-15288:2008 standard to obtain top-level requirements. Purpose of this process is "to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services" [6]. This can be accomplished in two phases: the *system requirements definition* and the subsequent *system requirements analysis and maintenance*. Figure 7 shows the two phases with their detailed steps. Similar to the process operation in Figure 4, it is possible to take iteration loops for optimizing the model.

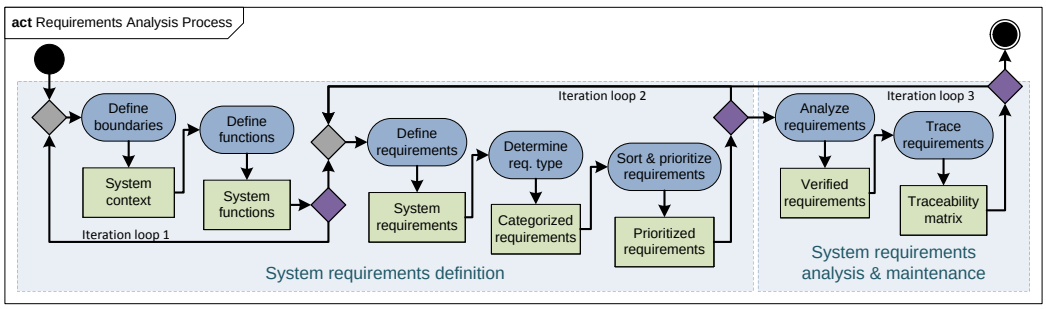

Figure 7 – Process for the model-based Requirements Analysis

Firstly, on a higher level of detail, the *system requirements definition* phase defines the system context. Based on the information captured in the *Stakeholder Requirements Definition Process* (cf. Figure 4) the system boundary is determined. Electrical and mechanical interfaces to external systems have to be examined and specified in negotiated Interface Control Documents (ICDs) [13]. In the project example the tangible user interface must have an interface to existing payment systems as those systems cannot be influenced or altered by the project. Required hardware has to fit into

the aircraft environment and has to operate with the available electrical power supply. Subsequently to the system context definition, the system functions have to be determined. These system functions should be independent from a solution or a design [13]. As those two steps for system context and functions are tightly corresponding with each other, it can be necessary to perform some iteration loops before proceeding with the next step (cf. Figure 7, iteration loop 1). Two exemplarily solution-independent functions of the tangible user interface are a payment and an authentication function.

In a next step it is now possible to define the system requirements. Like the stakeholder requirements they are also stored in the "Requirements" package at the system level of the model, but in a different sub-package to distinguish between both types of requirements [8]. System requirements can be categorized at least into two different types: *functional requirements* and *non-functional requirements*. While functional requirements describe the functions and features of a system, the non-functional requirements are constraining the system design [7, 10]. A functional requirement for the tangible user interface is for example: "Payment shall be conducted without additional interaction with the flight attendants". Non-functional requirements are directly derived from requirements belonging to stakeholder of the type "Law" and "Standard" as well as from solution constraints. For an NFC system in the aircraft cabin relevant standards like RTCA DO-160 [14], ISO/IEC-18092 [15] as well as many NFC-Forum standards [9] have to be included in the model as a large set of non-functional requirements. Sorting functional requirements helps to distinguish between mandatory and desirable requirements with facilitates the later architectural design.

Actions to be performed in the *system requirements analysis and maintenance* phase are very similar to those in the corresponding phase of the stakeholder requirements. By analyzing the set of requirements it should be checked that all of them are unique, complete, unambiguous, consistent and implementable [6]. Later it will be important that an implemented system can be validated against those verified system requirements. It is not conceivable for a system requirement to exist without having at least one superordinate stakeholder requirement. The other way round, stakeholder requirements should have at least one derived system requirement. A proof is done by using a tracing matrix mapping the dependencies between those requirements.

With completion of the *Requirements Analysis Process* all top-level requirements are comprehensively defined, analyzed, documented and maintained in the model and provide the basis for subsequent architectural design of the system (cf. Figure 2).

6 CONCLUSION

A methodology for modeling top-level requirements using the SysML and following approved ISO/IEC processes is introduced. The structured and targeted approach enables the definition of stakeholder and system requirements, even for complex systems where it is often difficult to define properly the overall context. With a consequent feature utilization of the SysML it is possible to improve the clarity of the model and to keep track of all model elements. By a structured capturing of information in a sin-

gle model repository, all data can be directly accessed and dynamically filtered. This model-centered requirements specification methodology is compatible with the subsequent model-based system design and implementation phases and enables consistency and perfect traceability for validation and verification during the overall MBRE process.

Acknowledgment

The work conducted in this study was supported by the LuFo V-1 project "Digitale und sichere Prozesse in der Kabine für den Passagier und die Besatzung (DiProPax!)", funded by the Federal Ministry for Economic Affairs and Energy based on the decision by the German Bundestag, which is gratefully acknowledged by the authors.

References

[1] C. Becker, T. Giese, D. Krakowski, S. Krittian, T. Scherer, "Efficiency of Model Based Methodologies in Air Systems Engineering", Proceedings of the 4th International Workshop on Aircraft System Technologies 2013, Hamburg, Germany, 81–96 (2013).

[2] Apple, "iPhone 6", www.apple.com/iphone-6/, visited: 2014/09/10 (2014).

[3] Federal Aviation Administration, "Press Release – FAA to Allow Airlines to Expand Use of Personal Electronics", Washington (2014).

[4] T. Kiehl, R. God, "An MBSE-Approach for using Near Field Communication in the Aircraft Cabin", Proceedings of the 4th International Workshop on Aircraft System Technologies 2013, Hamburg, Germany, 333–354 (2013).

[5] J.A. Estefan, "Survey of Model-Based Systems Engineering (MBSE) Methodologies", MBSE Initiative, (International Council on Systems Engineering, Seattle, 2008).

[6] ISO/IEC-15288:2008, "Systems and Software Engineering – System Life Cycle Processes", (International Organization for Standardization, Geneva, 2008).

[7] J. Holt, S.A. Perry, M. Brownsword, "Model-Based Requirements Engineering", (The Institution of Engineering and Technology, London, 2012).

[8] R. Karban, T. Weilkiens, R. Hauber, M. Zamparelli, R. Diekmann, A. Hein, "Cookbook for MBSE with SysML", (MBSE Initiative – SE^2 Challenge Team, 2011).

[9] NFC Forum, www.nfc-forum.org, visited: 2014/12/10 (2014).

[10] P. Bourque, R.E. Fairley, eds., "Guide to the Software Engineering Body of Knowledge", Version 3.0, www.swebok.org, (IEEE Computer Society, Washington, 2014).

[11] Lufthansa Technik AG, Software, "Kommunikationssystem und -verfahren in einem Flugzeug", patent application DE 10 2012 217 795 A1 (2014).

[12] Lufthansa Technik AG, "Kommunikationssystem in einem Flugzeug", patent application DE 10 2012 217 797 A1 (2014).

[13] INCOSE, "Systems engineering handbook: A guide for systems life cycle processes and activities", Version 3.2, (International Council on Systems Engineering, Seattle, 2010).

[14] RTCA DO-160G, "Environmental Conditions and Test Procedures for Airborne Equipment", (Radio Technical Commission for Aeronautics, Washington, 2010).

[15] ISO/IEC-18092:2013, "Information technology – NFC – Interface and Protocol (NFCIP-1)", (International Organization for Standardization, Geneva, 2013).