Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)

Model-based Safety Requirements Engineering for complex ATM Systems

Application in a Virtual ATC Tower Environment

Lothar Meyer, Michael Schultz, and Hartmut Fricke
Dresden University of Technology
Chair of Air Transport Technology and Logistics
Dresden, Germany
{meyer, schultz, fricke}@ifl.tu-dresden.de

Current design, development, and certification of air navigation systems become increasingly complex due to the contribution of many stakeholders to the development process. With increasing complexity, assuring safe operations in the ATM domain by determining safety requirements is harder to achieve and higher process integrity of all stakeholders is demanded. The proposed methodology and implementation into software shall support the mandatory safety assessment for certification issues by assuring a safe operation according to a certain target level of safety. The methodology augments the Eurocontrol safety assessment methodology by using a model-based approach, logic networks and linear algebra. Beside safety requirements, general system requirements can change during development phases and have to comply with numerous constraints and with economic criteria. The presented tool enables the user to evaluate the safety requirements by given criteria in short evaluation cycles to assure cost-optimized safety requirements for the system design. A cost function is presented that quantifies the achieved safety while also considering the economy of the chosen safety requirements. The methodology is finally applied to a safety assessment for the design of innovative virtual control tower ATC applications, which is performed with German ANSP Deutsche Flugsicherung. We could significantly improve the apportionment method for determining safety requirements.

Keywords: system safety assessment; functional hazard assessment; preliminary system safety assessment; risk assessment; design evaluation; technical requirements; human machine interfaces; ATM application

I. INTRODUCTION AND STATE-OF-THE-ART

Safety assessment of new but also existing air navigation systems (ANS) is a crucial and often demanding step to reach operational readiness and certification. It generally aims at formulating safety requirements for technical equipment, both for hard- and software components (RTCA DO-254 and RTCA DO-178B/248B) and to identify hazardous events or constellations potentially hampering ATM safety. The development of ANS typically comprises several safety critical tasks and functions leading to complex systems. Consequently, various experts such as product managers and system developers have to collaborate over a long period of system development, which often comprises various iteration cycles. Design functions and elements of ANS become hierarchic by allocating dedicated levels of criticality to them. The safe system operation is formally stated, if the ANS matches all set safety requirements.

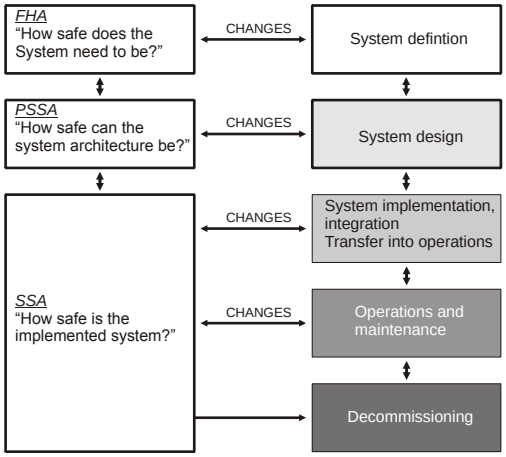

Figure 1. Development process dependencies between Safety Assessment and the Overall System Life Cycle [3].

According to Eurocontrol SAM [3], the first steps FHA and PSSA are realized during system definition (figure 1) and system design phase. They are performed usually by a heterogeneous working group. Starting with the identification of hazard causes, one can recognize a diversity of stakeholders and domains, each one having a distinct contribution to the probability of hazard occurrences. An involvement of all participants in the identification of causal events is therefore crucial and even mandatory. A commonly used approach is to perform workshops with all stakeholders to classify hazard causes with fault tree analysis models (FTA) to deliver a causal logic representation of occurrence. Safety requirements are then derived by workshop participants through apportionment [1] of safety objectives backwards into the FTA as depicted in figure 2. These recommended steps may easily be applied to operational systems, or to those for which at least all system design

[1]According to Eurocontrol SAM [4] terminology, the apportionment of safety objectives describes the activity of apportion given probabilities with respect to a causal logic and allocating the resulting probability to the causes.

requirements are final. However, the apportionment scheme between safety objectives resulting from the FHA and the safety requirement to be set in the PSSA relies on individual judgment and is as such uncertain. This uncertainty may in turn invoke misleading system design requirements allocating system costs that do not deliver additional safety.

This paper takes its motivation from here: It presents a standardized methodology on how apportioning safety objectives (SO) into safety requirements (SR) following minimization strategies.

The PSSA according to Eurocontrol [5], has the main objective to determine the appropriate design assurance level for hardware and software as e.g. avionics and ground equipment [24], communication and HMI [7] or HMI design issues [2]. Therein, safety requirements describe maximum acceptable probabilities that address causes of failure rates in the technical equipment or cognitive errors, faults and lapses of human operators.

When applying this apportionment methodology, the following effects occur:

- Requirements can change during development processes within the product development lifecycle. With respect to a living document character of the design requirements document in development processes, PSSA becomes a rather continuous process, which permanently updates hazard causes and causal logic according to changes of technical and functional requirements.
- The apportionment of safety objectives into safety requirements shall be applied by means of fault tree models. In contrast, the often experienced case has got the character of a logic gate network model, which superposes multiple fault trees. It results in a multi-relationship between cause probabilities and hazard probabilities. The typical process then is to ignore this multi-relationship leading to the random allocation scheme of various safety objectives to one hazard cause. An over-determination of safety requirements may result. An ordinary solution is the allocation of the lowest and strictest safety requirement. The disadvantage is a lowered economic system design due to the setting of non-optimized safety requirements thus exceeding the SO.
- Efforts resulting from increased reliability of system functions are often not taken into account during safety assessment. The common method of SO apportionment is a homogenous weighting along all hazard causes. Each cause is so treated equally contributing and consequently disregards a potential offset in the costs induced into the system design.

Consequently, the concept presented here aims at allocating the SOs such as to reach maximum cost efficiency. A detailed literature review reveals the following activities in this field, mainly since 1996, along with the publication of the [21] standards:

A tool for supporting software development processes was developed in [9], helping the user to automatically generate safety models in the UML modeling language and to optimize software architecture accordingly.

Developing UML system models form a basis to estimate hazards causes in hardware and software. Generating fault tree models from UML models was developed and investigated in [22].

Development processes contain numerous forms of UML-diagrams for organization and requirements purposes. The ability to interpret safety cases in these diagrams is supported by the Software product ‘Safety In The Loop’ [23].

The approach proposed here is to demonstrate a method and the implementation in a software tool that supports the system development process by intelligent and all-embracing apportionment of safety objectives while considering multi-relationships of hazard causes. Further, objectives of the method and the support tool are derived of the mentioned disadvantages of recommended apportionment method.

The functional requirements of the software implemented support tool shall provide a safety modeling function, in which the user is able

- to edit and delete hazard causes,
- to associate them causally with predefined hazards,
- to visualize causal logic with the help of fault trees and event trees (the combination of both diagrams is shown in Figure 2.
- to set quantitative safety requirements according to pre-set criteria,
- to verify correct apportionment of safety objectives to safety requirements, and
- to analyze the sensitivity between the achievable system design safety and the number and type of hazard causes known.

The following sections will present the methodology that refers to logic networks and extends the fault tree technique to support multi dependencies of hazard causes. A general transfer function of hazard causes to hazard events is deduced, which offers the ability to determine hazard occurrence depending on safety requirement settings. Two criteria for safety requirement determination are presented.

The used software development framework EMF (Eclipse Modeling Framework, see section III) is being used to implement and apply the method to ANS system relating to a novel working environment for ATC virtual tower operations[2]. The implementation is then used to apply PSSA safety objective allocation concept to the safety requirements.

[2] Research sponsored by the German Ministry of Economy and Industry, BMWI as subcontractor to Deutsche Flugsicherung GmbH (DFS).

II. METHODOLOGY

A. *Eurocontrol safety assessment methodology*

Recommendations on how to perform a system safety assessment for ANS certification issues may be found in [21] as discussed above. Eurocontrol Safety Assessment Methodology (EATMP SAM) [3] is the mostly agreed methodology for European safety Assessment in ATM which is mostly congruent to SAE ARP.

Therein, the functional hazard assessment (FHA) begins with the identification of functional hazards and the analysis of its consequences.

- In this context a hazard (H) definition determined by the Eurocontrol Safety Regulatory Requirement (ESARR 4) [6] reads as follows: "Any condition, event, or circumstances which could induce an accident" or defined by [3] as "anything that might negatively influence safety".
- A safety objective [6] is a qualitative or quantitative statement that defines the maximum frequency or probability at which a hazard can be tolerable (acceptable) to occur (Safety Target). They are derived both from the severity of the hazard effects and from the maximum probability of the effects, i.e. according the risk definition.
- A target level of safety [6] is the level of how far safety is to be pursued in a given context.

The consequence shall be assessed by a maximum number of occurrences (e.g. the target level of safety for accidents is 10^{-9} per operating hour [6]). Hazard events are then allocated with safety objective by calculating event tree branch probabilities with regard to these target levels of safety.

Following the FHA, a preliminary system safety assessment (PSSA) is performed to identify causal events, allocating safety requirements to them or extending the system design by applying risk mitigation strategies.

- Mitigation means [6] are any kind of internal means (people, procedures, and/or equipment) taken to control or prevent a hazard from causing events and to reduce risk of expected effects to a tolerable (acceptable) level.
- Safety requirements [6] are risk mitigation means, defined from the risk mitigation strategy, to comply with safety objectives. Safety requirements may take various forms, including organizational, operational, procedural, functional, performance, and interoperability requirements or environment characteristics.

The bow-tie diagram (figure 2) points out the link between hazard causes and hazard consequences. The definition of hazard consequences is part the FHA and the Event Tree Analysis (ETA) is the commonly used evaluation method. The Fault Tree Analysis (ETA) as part of the PSSA allows for an efficient mapping of the system design to derive the hazard causes.

Finally, the result of the system safety assessment (SSA) shall confirm whether the set safety objectives are met by the new system.

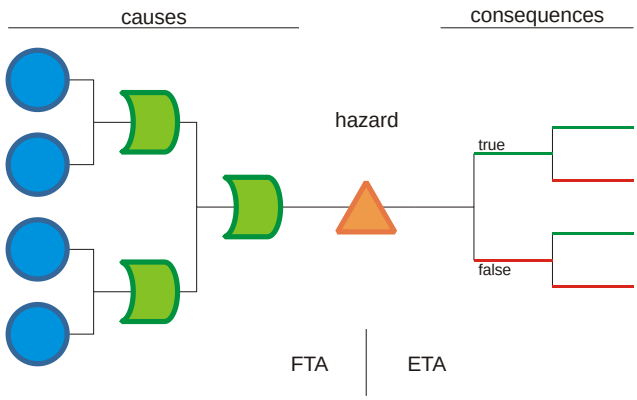

Figure 2. Bow-Tie Diagram linking FTA and ETA.

B. *Deriving hazard probabilites from cause probabilities*

The presented method of this section shows the derivation of a statistical transfer function, that supports the description of linked effects statistically and by means of a causal logic network. Causally forward, the transfer function determines hazard probability by a set of independent causative probabilities. Logic relationships between causes and hazards, expressed by means of 'and' and 'or' logic gates, are represented by network parameter values.

Having completed the FHA, safety objectives are available for every set of hazards that might impact the safety of operation with a fixed severity level. The transfer function shall allow deducing safety requirements out of this set of given safety objectives.

The approach for deriving the transfer function is complemented by the help of the axioms of unification and intersection by Kolmogorov [10]. Accordingly, the probability function $P(X)$ maps dependencies on an event X within a sample space.

Figure 3 illustrates a simple relationship between a hazard and its two causes, which can initiate the hazard occurrence independently. Assumed $P(H_1)$ as being the probability of hazard occurrence, the unification (OR) describes it depending on the causal probabilities with each one potentially initiating the hazard event independently:

$$P(H_1) = P(C_1 \cup C_2) = P(C_1) + P(C_2) - P(C_1 \cap C_2) \quad (1)$$

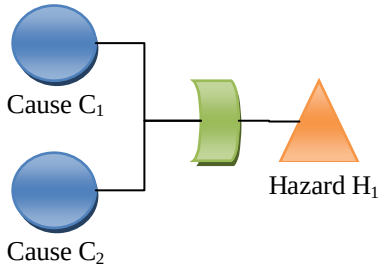

Figure 3. Simple causal relationship between a hazard and two primary causal events C_1 and C_2.

In addition, the equation of the type intersection (AND) describes the hazard probability with each cause potentially initiating the hazard event depending on one or more events:

$$P(H_2) = P(C_1 \cap C_2) \tag{2}$$

The composition of unification and intersection logic gates forms the logic network shown in figure 4.

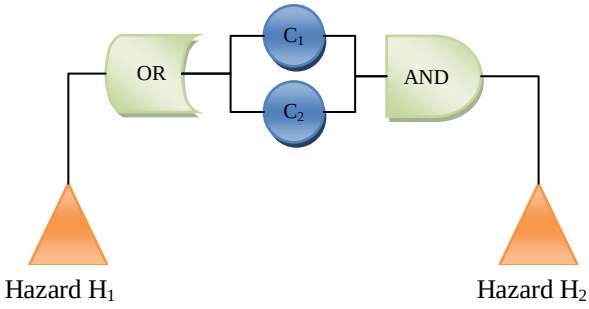

Figure 4. A Sample of a logic network composed of two fault trees.

An equation system can be formed of (1) and (2) by means of linear algebra to describe the probability relationship between causes and hazards as

$$\begin{bmatrix} P(H_1) \\ P(H_2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P(C_1) \\ P(C_2) \\ P(C_1 \cap C_2) \end{bmatrix} \tag{3}$$

including the nonlinear term $P(C_1 \cap C_2) = P(C_1) \cdot P(C_2)$. By generalizing this equation with the definition of the vector [3] $\vec{H} = [P(H_1),\ P(H_2),\ \cdots,\ P(H_n)]^T$, one can substitute (3) to

$$\vec{\mathrm{H}} = \underline{A} \cdot \psi(\vec{X}). \tag{4}$$

The equation includes the logic network matrix $\underline{A} \in \mathbb{R}^{n \times m}$, the vector representing all included probabilities of primary causes $\vec{X} = [P(C_1),\ P(C_2),\ \cdots,\ P(C_k)]^T$ and the combination function $\psi: \mathbb{R}^n \to \mathbb{R}^m$, $\psi(\vec{X})$ that combines causal probabilities without repetition. Depending on the number of independent causes, ψ includes products with combined factors of causal probabilities as nonlinear terms. Using (2) for demonstration purposes, the chosen strategy for defining ψ is

$$P(\mathrm{C}_1 \cap C_2) = P(C_1) \cdot P(C_2) = 10^{p_{C1} + p_{C2}} = 10^{\underline{B} \cdot [p_{C1} \quad p_{C2}]^T} \tag{5}$$

with $p_{C1} = \lg P(C_1)$ and $p_{C2} = \lg P(C_2)$ and $\underline{B} = [1 \quad 1]$. The exponent now offers a linear relationship to parameters that depends on the base 10 logarithm of (primary) causal probabilities.

Defining $\xi(\vec{v}) := 10^{diag(\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_\mathrm{i}) \cdot \vec{1}_\mathrm{i}}, \vec{v} \epsilon \mathbb{R}^i$ with the all-one vector $\vec{1}_i$ of the dimension i offers the possibility to augment the mentioned exponentiation operation for setting vector arguments and not to violate the limits and conventions of linear algebra. The approach is congruent for $\zeta(\vec{v}) := \lg diag(\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_\mathrm{i}) \cdot \vec{1}_i, \vec{v} \epsilon \mathbb{R}^i$. Expressing the exponent of (5) in linear matrix operations and $\vec{X} = \zeta(\vec{\tau})$ or rather $\vec{\tau} = \xi(\vec{X})$ one can define

$$\psi(\vec{X}) := \zeta\left(\underline{B} \cdot \xi(\vec{X})\right) \tag{6}$$

in which the linear multiplication with $\underline{B} \in \mathbb{R}^{m \times k}$ effects the combination without repetition of causal probabilities. In case of applying (6) to the example, $\underline{B}$ would result in

$$\underline{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}^T \tag{7}$$

The final nonlinear system is achieved by combining (4) and (6) and describes the transfer function from the primary causal probability vector to the safety objectives vector.

C. *Verification of compliance to safety objectives*

Safety requirements comply to given safety objectives $\overrightarrow{\mathrm{SO}} = [SO_1,\ SO_2,\ \cdots,\ SO_n]^T$ if

$$\underline{A} \cdot \zeta\left(\underline{B} \cdot \xi(\vec{X})\right) = \vec{\mathrm{H}} \leq \overrightarrow{\mathrm{SO}} \tag{8}$$

is satisfied. The solution space of compliant safety requirements is shown in figure 5 according to the sample of figure 4 with exemplarily chosen Safety objectives: $SO_1 = 10^{-3}\ h^{-1}$ and $SO_2 = 10^{-10} h^{-1}$.

[3] A vector is defined as n tuples of positive finite entries in a column matrix.

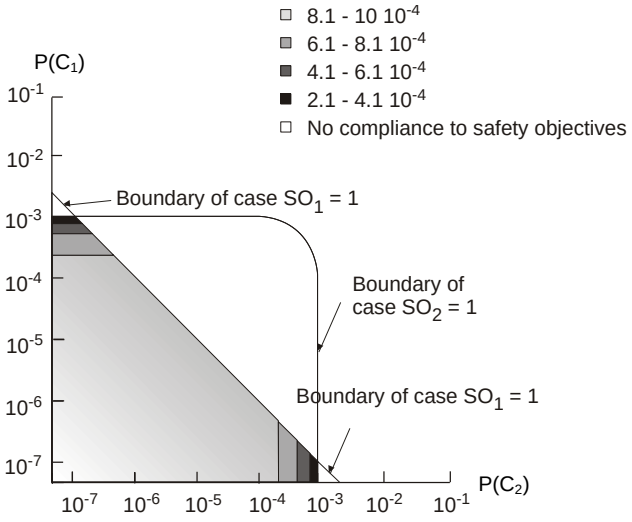

Figure 5. The solution space of compliant safety requirements is grey coded by the degree of exceeding safety objectives.

The resulting solution space is the intersection set of two sets. The first complies with SO_1 and the second with SO_2 (indicated by labeled lines in figure 5). In case of crossing solution boundaries, solutions for complying with all set safety objectives do exist.

D. Determining safety requirements

The set of compliant safety requirements forms a solution space. The design team can now choose inside that space to design hardware and software efficiently. A criterion to obtain a solution for determination purposes consists in setting boundary conditions that complement equation (7). The method is to set a ratio between two or more safety requirements. This ratio can be represented by a quotient of dividing one safety requirement with another.

$$\mathrm{P(C_1)} = \mathrm{y} \cdot \mathrm{P(C_2)} \tag{9}$$

The effect is a reduction of the system order (8) by one per condition. Consequently the under determined system becomes determined. The setting of a ratio to the sample (figure 5) results in an axis of tolerable safety requirements.

The available set of solutions offers the possibility of setting safety requirement according to economic criteria while granting safety. One criterion can be to assess solutions for the degree of exceeding the target probabilities given by the safety objectives. Applying the distance L_1 to the transposed (7) forms the safety cost function

$$J = \left\|\overrightarrow{\mathrm{SO}} - \vec{\mathrm{H}}\right\|_1 = \sum_{j=1}^{n} \left|\mathrm{SO_j} - \mathrm{H_j}\right|. \tag{10}$$

J becomes greater with decreasing H_j which consequently indicates a certain economic degree for the system design and the ability to quantify the achieved level of safety. The numerical result of that function (10) is also shown in figure 5. Assuming that a stricter safety requirement induces linearly a higher effort of ANS development, J can be interpreted as an economic degree with the black part as the best solution whereas the middle inner part represents the safest solution. For optimization purposes, the safety requirements that correspond to the safety cost minimum $J = 0$ can be determined as

$$\mathrm{J} = 0 \rightarrow \vec{X}_1 \cong \begin{bmatrix} 10^{-3} \\ 10^{-7} \end{bmatrix}, \vec{X}_2 \cong \begin{bmatrix} 10^{-7} \\ 10^{-3} \end{bmatrix}. \tag{11}$$

The analytic solutions are

$$\vec{X}_1 \cong \begin{bmatrix} 9.9\ 10^{-4} \\ 1.0\ 10^{-7} \end{bmatrix}, \ \vec{X}_2 \cong \begin{bmatrix} 1.0\ 10^{-7} \\ 9.9\ 10^{-4} \end{bmatrix} \tag{12}$$

which offer more precise solutions due to the resolution limits of numerical calculation.

III. IMPLEMENTATION

The common model-based development approach refers to the use of domain-specific characteristics to analyze the requested system behavior. The model presented here allows for system evaluation at an early development stage and finally provides a solid structure for (automated) code generation and testing. The typically observed split of system designers and safety engineers often leads to misunderstanding and significantly lowers the efficiency of the development process. The model based safety analysis offers the opportunity to use conceptual models for early safety analysis. The scope of our approach is to standardize the ETA generation by an appropriate software environment and provide a reliable link between the evaluated FHA hazards and the conceptual (functional) model according to figure 2. Inside the conceptual model this connection allows for an automatic calculation.

Such tool support will help the safety engineer to focus on his primary task instead of spending time for gathering architectural details about the system behavior [11], [12]. Following the intention of model-based design, formal safety models could be directly integrated in this scheme [14], [17], [18], [19]. Since safety assessment requires significant knowledge of the system functions and behavior, an efficient connection seems to be valuable. Besides the improved performance of safety analysis, a positive economic side effect of the common model base could save cost on both the development and safety process. Further on, this extended system model can be utilized for theorem proving, consistency analysis or model checking [15], [16], [20].

To realize the model-based safety analysis an adequate software environment has to be designed and implemented. There are several tools and integrated development environments (IDE) to develop and evaluate models. Usually, the IDEs are consists of: code editor, compiler/interpreter, building and debugging tools. We choose the Eclipse Modeling Framework (EMF), because this framework provides an abstract infrastructure with the capability of modeling transformation and evaluation on an open-source basis. In compliance to [1] we use TOPCASED (Toolkit in Open source for critical applications and systems

development) which targets aerospace systems [25], [26], [27]. We choose Java as programming language for our software environment. The EMF consists of three major components: the Meta Model `Ecore`), derived from Unified Modeling Language - UML), the `EMF.edit` framework for reusable classes for building editors and the `EMF.gen` code generator including a graphical user interface. For a safety assessment environment, each of the existing elements has to be modeled as an `Ecore` representation. The following figure 6 shows a simplified graphical representation of the associated `Ecore` model.

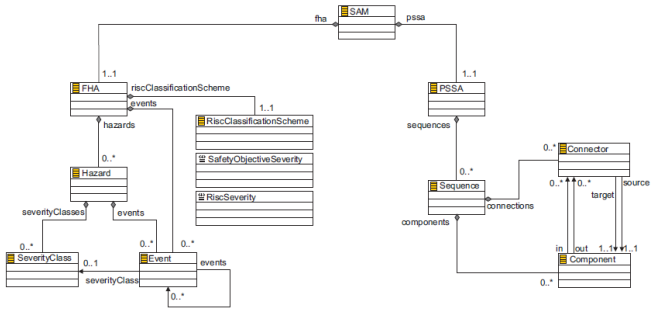

Figure 6. Simplified SAM model as Ecore representation [28]

On the basis of the `Ecore` model, a transfer to a `GenModel` allows for automatic code generation. The automatic generation is considered very important, since it may significantly reduce transfer errors. So, the system developer can focus on his main modeling task. In addition to the EMF, the Graphical Modeling Framework (GMF) provides a set of generic components and runtime infrastructures for graphical editors. The generated graphical environment provides the complete functionality of the Eclipse Framework associated with the graphical capabilities of the GMF. The user is able to initiate the safety analysis, where he can choose both, the Eclipse standard feature of editing the `Ecore` representation (provided by the EMF functionality) or the advanced graphical representation of the corresponding GMF environment (figure 7).

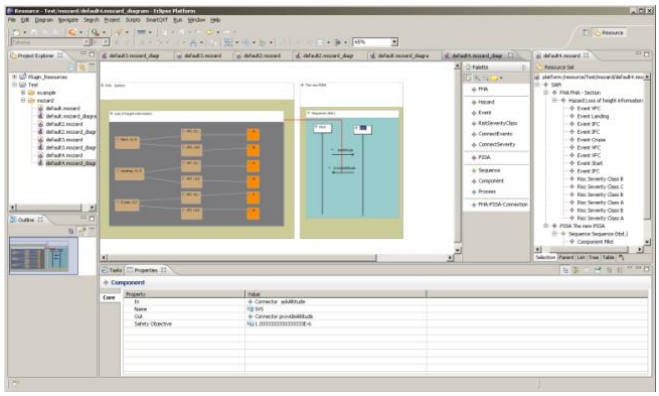

Figure 7. A SAM environment based on TOPCASED [28]

IV. Demonstration

The virtual control tower project VICTOR[4] is part of a re-design of the tower controller working environment and is currently under safety investigation. A major component of the new tower system is the substitution of the out of window view by a vision system comprising e.g. video cameras and synthetic vision functions to provide visual support to the aerodrome controller. Important safety issues of the virtual tower vision system are e.g. the contributions of the HMI design to the safe perception of presence, position of aircrafts and the presence of wildlife on the airfield. According to the performed FHA [2], these data demands are used for safety critical decisions by the controller and effect most severe consequences (up to accidents) in case of defective perception. Thus, each of the mentioned hazards are allocated with a safety objective complying with ESARR4.

As a part of the PSSA the hazard causes identification has been performed experimentally. Causative parameter values, forming part of the video resolution, contrast and object size could be identified as contributing essentially to the probability of each mentioned hazard occurrence. The fault tree (figure 8) results by assessing experimental output data and the related test person statements.

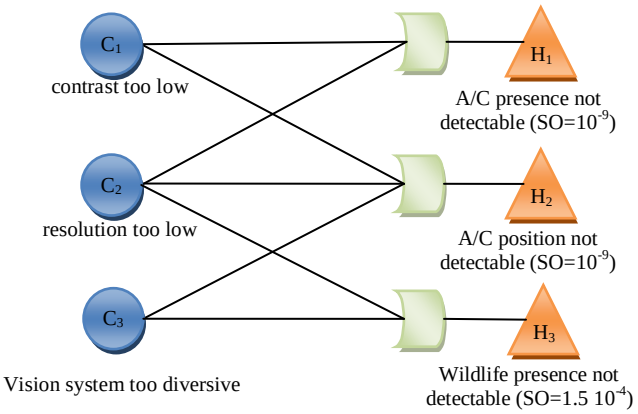

Figure 8. A fault tree sample of the experimental identification of causative events.

Basing on figure 8, the logic network matrix results as

$$\underline{A} = \begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 0 & 1 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}. \qquad (13)$$

Figure 9 shows the space of solutions for compliant safety requirements.

[4] The Virtual Control Tower Research Study VICTOR is part of the forth aviation research program LuFo of the German Federal Ministry of Economics and Technology and provided by the German ANSP Deutsche Flugsicherung GmbH.

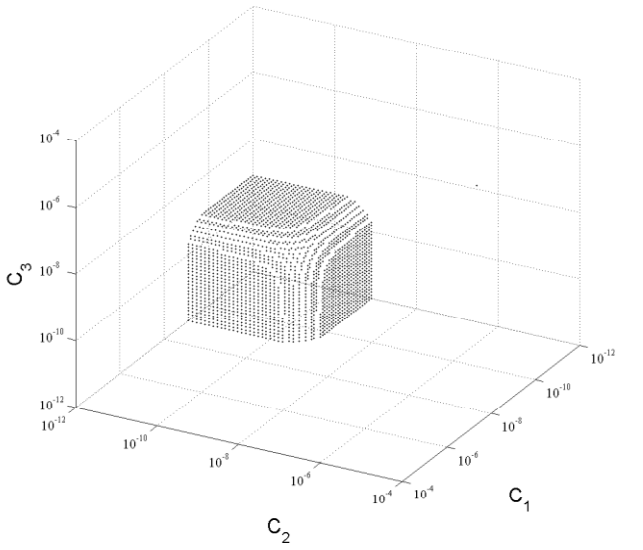

Figure 9. Boundaries of three safety requirements that are compliant to safety objectives

According to the safety cost criterion (10), there is a homogenous safety cost of J=1.5 10^{-4} for any bounding solution. Consequently, this case contains bounding solutions that are of equal economic value. For this reason, the simplest approach of determining the safety requirements is to set boundary conditions that force a homogenous result.

$$P(C_1) = P(C_2) = P(C_3) = 3.16\ 10^{-10}\ \text{h}^{-1} \quad (14)$$

This approach neither respects the technical complexity of a dedicated hardware nor respect development expenditures. It rather demonstrates the ability to evaluate hardware and software safety by the use of this method and the potential extension by safety cost criteria. The consequent transfer of the mathematical methods into the developed environment allows immediately for using our results in associated research projects (figure 10).

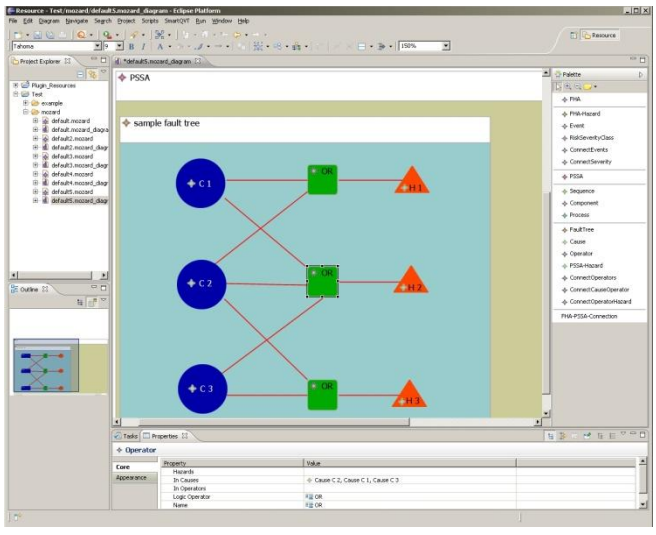

Figure 10. Transfer of mathmatical model into the developed environment

V. CONCLUSION

The effort of developing methods to support design processes is to offer a new level of efficiency and reliability in designing and configuring system components with respect to the stringent safety requirements which need to be fulfilled along the certification process of ATM systems. A method to verify the compliance of safety requirements to safety objects that respects logic networks has been developed. An evaluation has been performed by determining safety requirements to elemental design issues of the virtual control tower. This method relies on numerical determination of solution spaces, in which boundary conditions and safety cost functions can be set to locate optimal solutions. The effort of integrating system developer issues in safety analysis promises the ability for achieving a shorter safety evaluation cycles in system design processes.

The methodology is demonstrated with 2 and 3-D samples with cases that illustrate solution spaces of probabilities of independent cause occurrences. Solution criteria may be set by selecting target costs e.g. exceeding hazard probabilities or by defining safety requirement rations for the degradation of the system order. A solution that represents safety compliant requirements becomes determinable.

The exemplary cases were of course trivial and not representative for complex air navigation system architectures that include more safety-relevant causes. Our future efforts will concentrate on applying and evaluating the demonstrated methodology on further research and development cases such as the ongoing virtual tower system design. Thereby, the implication of a weighting that represents expenditures of system development to achieve safety compliance is beneficial for cost management and will be focused.

A major clue of the implementation is the summarizing view on all causal events and elements that compromises the safe operation of the system. By the use of an all-including logic network and by integration into a safety management system, all stakeholders of an ANS would be able to contribute to a causal management data base and thus be able to assure that safety objectives will be matched.

REFERENCES

[1] B. Langer (2010), *Bridging the gap between users and developers with model-based usage analysis*, DGLR.

[2] L. Meyer et al. (2010), *Functional Hazard Analysis of Virtual Control Towers*, Valenciennes, IFAC.

[3] SAM-TF (2004), *Air navigation system safety assessment methodology*, Eurocontrol, Brussels, Belgium.

[4] SAM-TF (2004), *Preliminary system safety assessment*, Eurocontrol, Brussels, Belgium.

[5] SAM-TF (2004), *Preliminary system safety assessment - guidance material A: safety requirements*, Eurocontrol, Brussels, Belgium.

[6] ESARR SRC (2001), *Eurocontrol safety regulatory requirement 4 - risk assessment and mitigation in ATM*, Eurocontrol, Brussels, Belgium.

[7] EATMP (2006), *A-SMGCS Levels 1 & 2 Preliminary Safety Case*, Eurocontrol, Brussels, Belgium.

[8] M. Schultz, B. Langer, L. Meyer and H. Fricke (2010), *Model-Based Safety Analysis Considering ATM Domain Requirements*, Dresden, Germany.

[9] J. Briones and M. de Miguel (2006), *Integration of safety analysis and software development methods* , Madrid, Spain.

[10] A. Kolmogorov (1933), *Grundbegriffe der Wahrscheinlichkeits-rechnung*, Berlin, Germany.

[11] A. Joshi et al. (2006), *Model-Based Safety Analysis Final Report, NASA contractor report*, CR-2006-213953.

[12] D.J. Pumfrey, (1999), *The principled design of computer system safety analyses*, Ph.D. thesis, University of York.

[13] O. Lisagor et al. (2006), *Towards a Practicable Process for Automated Safety Analysis*, ISSC, pp. 596-607.

[14] M. Güdemann et al. (2010), *Probabilistic Model-Based Safety Analysis*, EPTCS 28, pp. 114 -128.

[15] S. Miller et al. (2006), *Proving the Shalls - Early validation of requirements through formal methods*, STTT, 8(4-5), pp. 303-319.

[16] M.W. Whalen et al. (2008), *Integration of Formal Analysis into a Model-Based Software Development Process*, MICS 2007, LNCS 4916, pp. 68-84.

[17] O. Lisagor, J.A. McDermid, and D.J. Pumfrey (2006), *Towards a Practicable Process for Automated Safety Analysis*, ISSC, pp. 596607.

[18] D. Domis and M. Trapp (2008), *Integrating Safety Analyses and Component-Based Design SAFECOMP*, LNCS 5219, pp. 5871.

[19] M. Bretschneider, H.-J. Holberg, E. B¨ ode, I. Brckner, T. Peikenkamp,and H. Spenke (2004), *Model-based Safety Analysis of a Flap Control System*, INCOSE.

[20] M. Bozzano, A. Cavallo, M. Cifaldi, L. Valacca, and A. Villafiorita (2003), *Improving Safety Assessment of Complex Systems : An Industrial Case Study*, FM, pp. 208-222.

[21] SAE (1996), *ARP 4754 - certification considerations for highly-integrated or complex aircraft systems*, Warrendale, USA.

[22] H.J. Pai, J.B. Dugan (2002), *Automatic Synthesis of Dynamic Fault Trees from UML System Models. In: International Symposium on Software Reliable Engineering*, IEEE Computer Society, Los Alamitos.

[23] F. Thom (2002), *Safety In The Loop: An Overview Of The System Safety Issue Throughout The Product Development Lifecycle*, ARTiSAN Software Tools Ltd, Cheltenham, UK.

[24] J. Hammer, G. Caligaris, M. Llobet (2007), *Safety analysis methodology for ADS-B based surveillance applications*, ATM Seminar 2007, Barcelona, Spain.

[25] A.P. Gaufillet and B.S. Gabel (2010), *Avionic Software Development with TOPCASED SAM*, ERTS.

[26] P. Farail et al. (2006), *The TOPCASED project*, ERTS.

[27] A.R. Faudou et al. (2010), *TOPCASED Requirement: a model-driven, open-source and generic solution to manage requirement traceability*, ERTS.

[28] M. Schultz, L. Meyer, B. Langer, and H.Fricke (2011), *Model-based Safety Assessment as Integrated Part of System Development*, Workshop on Aviation System Technology.

AUTHORS BIOGRAPHY

Lothar Meyer (Hannover, Germany, 1981) studied electrical engineering at Technische Universität Dresden (TUD) from 2002-2008. During internships at Airbus Deutschland GmbH and VW AG he gained experiences in the field of automation and closed loop control. After graduating he began at the chair of Air Transport and Technology and Logistics at TUD as scientific assistant. The activities include lecturing (aerodynamics, qualities of flight and flight control) and the mentoring of a safety assessment in scope of the research project Virtual Control Tower Research Study (VICTOR) provided by the ANSP Deutsche Flugsicherung GmbH.

Michael Schultz (Rostock, Germany, 1976) studied business and engineering at TUD from 1996-2002. During several internships at Siemens Financial Services and the BMW Research Center he gained experiences in the field of quality engineering and system design. After 2 years at automotive industry as a system and quality engineer, he changed to the chair of Air Transport Technology and Logistics at TUD. Since the end of 2003 he works as a scientific assistant and lecturer (aerodynamics, flight mechanics, and terminal operations). He is in charge of several research projects related to managing complex passenger dynamics, safety assessment, stochastic modeling of agent-based systems and turnaround optimization. In 2010 he finished his PhD on logistics and aviation focused on the individual-based model for passenger movement behavior.

Hartmut Fricke (born in Berlin, Germany in 1967) studied Aeronautics and Astronautics at Technische Universität (TU) Berlin from 1985-1991. From 1991 to 1995 he was a research fellow in Flight Operations, Airport Planning, and ATM at TU Berlin, where he completed his doctor thesis in ATM (ATC-ATFM Interface). In 2001 he finished his Habilitation on “Integrated Collision Risk Modeling for airborne and ground based systems”. This included HIL experiments with an A340 full flight simulator in co-operation with EUROCONTROL Experimental Center (EEC). Since December 2001 he has been Head of the Institute of Logistics and Aviation, and Professor for Aviation Technologies and Logistics at TU Dresden. In 2006 he was appointed Member of the Scientific Advisory “Board of Advisors” to the Federal Minister of Transport, Building and Urban Affairs in Germany.