Finite-state machine method in the safety assessment process using Stateflow diagrams

A S Savelev¹, E V Eroshchenkov¹, E S Neretin¹ and D A Shevela²

¹ Moscow Aviation Institute (National Research University), 4, Volokolamskoe shosse, Moscow, 125993, Russia
² Bauman Moscow State Technical University, 5, ul. Baumanskaya 2-ya, Moscow, 105005, Russia
E-mail: arssavelev@mai.education

Abstract. Safety Assessment Process is one of the most important processes of the avionics systems design. Modern and perspective aircraft uses a Model-Based Design approach, but Safety Assessment performing with a high risk of error due to human factors. Nowadays design tools, like MATLAB Simulink and ANSYS medini analyze, propose new methods of the safety assessment process—Model-Based Safety Assessment (MBSA). The types of MBSA were analyzed and classified. This work describes the MBSA approach in general and focuses on MBSA in part of the failure modeling using the finite-state machine method. The use of this method can help both the safety team and development team to have a clear dialogue about the system’s work and failures.

1. Introduction

Figure 1 describes a top-level view of the safety assessment process (Functional Hazard Assessment, Preliminary System Safety Assessment, and System Safety Assessment) and how the safety assessment methods relate to that process. The development process is iterative in nature. The safety assessment process is an inherent part of this process. The safety assessment process begins with the concept design and derives the safety requirements for it. As the design evolves, changes are made and the modified design must be reassessed. This reassessment may create new derived design requirements. These new requirements may necessitate further design changes. The safety assessment process ends with the verification that the design meets the safety requirements. A typical development cycle timeline is shown across the top to show the chronological relationship of the safety process to the development process. Safety processes that are linked in the design process are grouped in boxes to highlight this relationship.

A Functional Hazard Assessment (FHA) is conducted at the beginning of the aircraft/system development cycle. It should identify and classify the failure condition(s) associated with the aircraft functions and combinations of aircraft functions. These failure condition classifications establish the safety objectives (like a required probability per flight hour) [1].

The PSSA is a systematic examination of the proposed system architecture(s) to determine how failures can cause the functional hazards identified by the FHA. The objective of the PSSA is to establish the safety requirements of the system and to determine that the proposed architecture can reasonably be expected to meet the safety objectives identified by the FHA. The PSSA is also an interactive process associated with the design definition. The PSSA
is conducted at multiple stages of the system development including system, item, and hardware/software design definitions. At the lowest level, the PSSA determines the safety-related design requirements of hardware and software. The PSSA usually takes the form of an FTA (DD or MA may also be used) and should also include common cause analysis.

The System Safety Assessment (SSA) is a systematic, comprehensive evaluation of the implemented system to show that safety objectives from the FHA and derived safety requirements from the PSSA are met. The SSA is usually based on the PSSA FTA (DD or

---

**Figure 1.** Overview of the classic System Safety Assessment Process.
MA may also be used) and uses the quantitative values obtained from the Failure Modes and Effects Summary (FMES). The SSA should verify that all significant effects identified in the FMES are considered for inclusion as primary events in the FTA. The FMES is a summary of failures identified by the FMEA(s) which are grouped based on their failure effects. The SSA must also include applicable common cause analysis results [1].

The main problem of carrying out the FHA is the lack of regulatory procedures and accepted processes for validating the classification of the requirements, according to the criticality of each failure condition. This problem can be solved with a Model-Based Design approach [2]. The main problem of carrying out the PSSA and SSA is the high probability of error due to the human error factor.

The research aims to improve the safety of the developed systems using the example of avionics systems. The proposed solutions are expected to shorten development time and minimize the risk of human factors through the use of MBSA. To ensure this aim of the study, the following tasks are solved:

- development of qualitative and quantitative methods for performing Common Cause Analysis. These methods will allow for the detection and prevention of potential common causes of failure during the design phase;
- development of a model of avionics systems to ensure validation and verification of safety requirements using a model-based approach;
- development of a means of automating the verification of failure design cases using algorithms for computer processing of graphic and sound information.

This paper focuses on the application of Model-Based Design practices to the Safety Assessment Process using a finite-state machine method.

2. Methodology

2.1. Model-Based Safety Assessment Principles

The model-based approach is widely used in the development of onboard software. For the aviation industry, DO-331 has been issued. However, the scope of this document does not apply to the use of MBSA products under development. This science is promising and actively developing. Many methodologies have been proposed for developing models for the SSA. They can be divided into two groups: creating extensions to existing high-level modeling languages and developing new languages. Within the framework of the first approach, one can distinguish:

- extending the development and analysis language of AADL architectures using the error model [3];
- use of reliability data in the development of models in MATLAB/Simulink with the subsequent generation of Fault Trees Analysis and Failure Mode and Effect Analysis [4];
- converting SysML Models to Fault Trees [5].

The second approach highlights:

- the SafeDeML modeling language for the safety assessment of complex systems (used in the automotive industry, where safety assessment processes are similar to those in the aviation industry) [6];
- the AltaRica 3.0 language is a version of the AltaRica language developed to perform safety assessment [7].

Demonstrated that MBSA is a great research topic. This work discusses the use of failure types and state changing in more detail. The results improve the models needed to validate and verify the safety requirements for avionics systems.
2.2. Finite state machine methodology

Finite state machines are representations of dynamical systems that move from one mode of state to another. Finite state machines:

- serve as a high-level starting point for the software development process;
- allow to focus on the modes of operation and conditions necessary for the transition from one mode to another;
- help in developing models that remain clear and concise, even as the complexity of the model increases [8].

Avionics systems design relies on state machines to control complex logic. Traditionally, developers have used truth tables to represent the relationships between inputs, outputs, and states of a state machine [9]. The resulting table describes the logic needed to investigate the behavior of the system.

Another approach to avionics systems design is to model the behavior of a system using Stateflow by describing it in terms of state transitions. Stateflow is a graphical programming environment based on state machines that are designed to test and debug the logic of the system’s functioning, as well as consider various modeling scenarios and generate code from a state machine [10].

A Stateflow diagram can contain sequential and combinatorial logic in the form of state transition diagrams, block diagrams, state transition tables, and truth tables. States and transitions form the basic building blocks of a sequential logic system. Another way to represent sequential logic is a state transition table, which allows you to enter state logic in table form [11].

2.3. Safety Assessment using Stateflow diagrams

The specified properties of the finite state machine method can be used in an MBSA using the example of the Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA). FMEA is used to study the consequences of single failures and to confirm that no single failure leads to a catastrophic situation. FTA is used to validate and verify quantitative safety requirements through graphical representations of causes of failure (from FMEA), connected by boolean operators [12].

![Diagram](image)

**Figure 2.** MBSA process for FMEA and FTA using Stateflow.

Stateflow provides operate-to-failure modeling of state transitions for FMEA purposes and can be configured to automate the generation of fault trees based on failure propagation logic. Figure 2 demonstrates this process.
3. Results and discussion

3.1. Study Case

For a simple example, we study an abstract device, which has three input interfaces and one output interface. Its means, that failure can be initiated by external signals or internal: everyone Input interface leads to a fault of one of the three channels; Computer and Output interface lead to three channels fault. Device architecture is presented in figures 3 and 4.

![Figure 3. A device like a black box.](image)

![Figure 4. A device like a clear box.](image)

For simplicity, we will only consider the failure modes of the “loss” type, without “malfunction”. In this case, equally, the device has three states, which may arise due to a single failure (required to perform FMEA):

1) failures are absent;
2) loss of one channel;
3) total loss.

Another state, which may arise in case of single failures combinations (required to perform FTA):

4) loss of two channels.

According to figure 2, these failure modes should be transmitted to the devise/system design team, which can perform a Stateflow diagram using finite state machines.

3.2. Results

Depending on the input data (failures of components), a Stateflow diagram was drawn up (figure 5).

![Table 1. FMEA example.](image)

| Failure mode                          | Effect          |
|---------------------------------------|-----------------|
| Loss of one Input interface unit      | Loss of one channel |
| Loss of Computer unit                 | Total loss      |
| Loss of Output interface unit         | Total loss      |

Results of this modeling should be used in the FMEA. Example of the qualitative FMEA worktable present in table 1.
4. Summary
The MBSA process using Stateflow was analyzed for avionics systems. This method can help to design and safety specialists in the interaction in questions about the system working and also allows to exclude human subjectivity when making decisions about the effects of failures since state space are limited and deterministic.

Future works in this area will include some directions:
• Automatic Fault Tree generation;
• Interactions with the tools needed for Safety Assessment (for example, ANSYS medini analyze);
• Adopting Stateflow models to perform validation of the Functional Hazard Assessment of the Avionics systems functions in the “real-time” modeling.

Acknowledgments
This work was supported by the Russian Foundation for Basic Research (Grant No. 20-31-90028).

References
[1] 1996 Aerospace Recommended Practice SAE ARP 4761 Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment (USA: SAE International) p 331 URL https://www.sae.org/standards/content/arp4761/
[2] Savelev A et al 2020 IOP Conf. Ser.: Mater. Sci. Eng. 868(012006) DOI: 10.1088/1757-899X/868/1/012006
[3] Brunel J et al 2017 Performing Safety Analyses with AADL and AltaRica Model-Based Safety and Assessment. IMBSA 2017. Lecture Notes in Computer Science vol 10437, ed M Bozzano and Y Papadopoulos (Cham: Springer) DOI: 10.1007/978-3-319-64119-5_5
[4] Shao N et al 2017 Model-based safety analysis of a control system using Simulink and Simscape extended models MATEC Web of Conferences 139(00219) DOI: 10.1051/matecconf/201713900219
[5] Munk P and Nordmann A 2020 Model-based safety assessment with SysML and component fault trees: application and lessons learned Softw. Syst. Model. 19 889–910 DOI: 10.1007/s10270-020-00782-w
[6] Gonschorek T, Bergt P, Filax M and Ortmeier F 2019 Integrating Safety Design Artifacts into System Development Models Using SafeDeML Model-Based Safety and Assessment. IMBSA 2019. Lecture Notes in Computer Science vol 11842, ed Y Papadopoulos, K Aslansefat, P Katsaros and M Bozzano (Cham: Springer) DOI: 10.1007/978-3-030-32872-6_7
[7] Batteux M, Prosvirnova T, Rauzy A and Kloul L 2013 The AltaRica 3.0 project for model-based safety assessment 11th IEEE International Conference on Industrial Informatics (INDIN) (Bochum) pp 741–746 DOI: 10.1109/INDIN.2013.6622976

[8] Stateflow®Getting Started Guide © COPYRIGHT 2004–2020 by The MathWorks, Inc. URL https://www.mathworks.com/help/pdf_doc/stateflow/stateflow_gs.pdf

[9] Dellen C and Liebner G 1991 Automated code generation from graphical, reusable templates IEEE/AIAA 10th Digital Avionics Systems Conference (Los Angeles, CA, USA) pp 299–304 DOI: 10.1109/DASC.1991.177183

[10] Manamcheri K, Mitra S, Bak S and Caccamo M 2011 A step towards verification and synthesis from simulink/stateflow models Proceedings of the 14th international conference on Hybrid systems: computation and control (HSCC’11) (New York, NY, USA: Association for Computing Machinery) pp 317–318 DOI: 10.1145/1967701.1967749

[11] Stateflow ®Reference © COPYRIGHT 2006–2020 by The MathWorks, Inc. URL https://www.mathworks.com/help/pdf_doc/stateflow/stateflow Ug.pdf

[12] Aerospace Recommended Practice SAE ARP 4761 1996 Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment (USA: SAE International) p 331 URL https://www.sae.org/standards/content/arp4761/