2013

Fault Tree Analysis for Safety/Security Verification in Aviation Software

Andrew J. Kornecki
Embry-Riddle Aeronautical University, kornecka@erau.edu

Mingye Liu
Embry-Riddle Aeronautical University

Electronics **2013**, *2*, 41-56; doi:10.3390/electronics2010041

OPEN ACCESS

electronics

ISSN 2079-9292

www.mdpi.com/journal/electronics

Technical Note

Fault Tree Analysis for Safety/Security Verification in Aviation Software

Andrew J. Kornecki * and Mingye Liu

ECSSE, Embry Riddle Aeronautical University, Daytona Beach, FL 32114, USA;
E-Mail: lium1@my.erau.edu

* Author to whom correspondence should be addressed; E-Mail: kornecka@erau.edu;
Tel.: +1-386-226-6888; Fax: +1-386-226-6678.

Received: 27 September 2012; in revised form: 15 January 2013 / Accepted: 21 January 2013 / Published: 31 January 2013

Abstract: The Next Generation Air Traffic Management system (NextGen) is a blueprint of the future National Airspace System. Supporting NextGen is a nation-wide Aviation Simulation Network (ASN), which allows integration of a variety of real-time simulations to facilitate development and validation of the NextGen software by simulating a wide range of operational scenarios. The ASN system is an environment, including both simulated and human-in-the-loop real-life components (pilots and air traffic controllers). Real Time Distributed Simulation (RTDS) developed at Embry Riddle Aeronautical University, a suite of applications providing low and medium fidelity en-route simulation capabilities, is one of the simulations contributing to the ASN. To support the interconnectivity with the ASN, we designed and implemented a dedicated gateway acting as an intermediary, providing logic for two-way communication and transfer messages between RTDS and ASN and storage for the exchanged data. It has been necessary to develop and analyze safety/security requirements for the gateway software based on analysis of system assets, hazards, threats and attacks related to ultimate real-life future implementation. Due to the nature of the system, the focus was placed on communication security and the related safety of the impacted aircraft in the simulation scenario. To support development of safety/security requirements, a well-established fault tree analysis technique was used. This fault tree model-based analysis, supported by a commercial tool, was a foundation to propose mitigations assuring the gateway system safety and security.

Keywords: national airspace system; simulation; safety; security; fault tree

1. Introduction

The early phases of the software development life cycle that include description of the concept, requirements specification and design descriptions are the foundation of the entire development project. The quality of the requirements and design documents has a significant impact on the remaining deliverables and the ultimate outcome of the project. Data in the NIST report [1] show that about 70% of defects are introduced in the requirements and design phases. It is quite costly to fix those defects due to the need of a substantial rework from the beginning and through almost all life-cycle phases. The cost increases five- to thirty-fold when the defects are corrected in the subsequent phases of the lifecycle. From this perspective, requirements are the top factor in a project's success or failure.

The safety and security of a system is of primary concern for dependable systems in regulated industries like aerospace, medical, nuclear, transportation, *etc.* One needs to thoroughly analyze the hazards due to not only system failures, but also external conditions caused by both benevolent and malicious events. The Fault Tree Analysis (FTA) technique is one of the most popular to be applied in such scenarios. We applied FTA to the analysis of a component of NextGen simulation.

Appropriate FTA models have been built to develop safety/security requirements, and the possible mitigation means have been proposed. The paper is organized as follows. Section 2 describes the application domain. In Sections 3 and 4, we introduce safety, security and fault tree analysis concepts. Section 5 describes the detail of the fault tree model. Section 6 describes simulation results and analyses using the selected tool. In section 7, we present the resulting safety and security requirements and propose mitigation methods.

The paper's main contribution is to show how application of the FTA technique leads to identification of safety and security requirements of the gateway and, subsequently, proposing appropriate mitigations.

2. Application Domain

The Next Generation Air Traffic Management (ATM) system (NextGen), to replace the contemporary ground radar-based system, is a blueprint of the future National Airspace System (NAS) based on satellite navigation and advanced system interconnectivity [2]. One of the NextGen products is the Enhanced Traffic Management System (ETMS) designed to react to air traffic congestion and, thus, improving the efficiency of the system. The long-term transition plan is implemented in stages, with the final implementation planned for 2025. The NextGen Global Positioning System (GPS) technology is designed to save time and fuel, reduce delays, increase capacity and permit controllers to monitor and manage aircraft with greater safety margins. In June 2010, American and European authorities reached a preliminary agreement on interoperability between their future air traffic management systems, NextGen and its European equivalent, SESAR (the Single European Sky ATM Research), supporting the Single European Sky legislation [3].

NextGen consists of five elements [2]:

- Automatic Dependent Surveillance-Broadcast (ADSB) will provide air traffic controllers and pilots with much more accurate information using Global Positioning System (GPS) satellite signals. Aircraft transponders receive GPS signals and use them to determine the aircraft's precise position in the sky and on the ground.
- System Wide Information Management (SWIM) will provide a single infrastructure and information management system to deliver data to many users and applications, by reducing the number and types of interfaces and systems.
- Next Generation Data Communications will provide exchange of routine controller-pilot messages and clearances via data links, reducing the need for extensive voice communications, improving air traffic controller productivity and enhancing capacity and safety.
- Next Generation Network Enabled Weather (NNEW) will provide a common weather picture across the entire national airspace system, fusing thousands of weather observations and sensor reports from ground-, airborne- and space-based sources.
- NAS Voice Switch (NVS), replacing seventeen different voice switching systems in the NAS with a single air/ground and ground/ground voice communications system.

The NextGen system is a collaborative project of industry, government and academic entities. It is evident that development of such a complex system requires a great deal of preparation. One of the critical questions to ask is: "would it work?" To answer such a question, one needs to simulate and evaluate a variety of scenarios reflecting potential situations that may occur in the national airspace. Aviation Simulation Network (ASN) allows collaboration within the aviation community, enabling integration of a variety of real-time simulations via the Internet.

Embry Riddle Aeronautical University (ERAU) is an academic partner in the NextGen Program. The NextGen ERAU Aviation Research Laboratory (NEAR Lab) is dedicated to support NextGen research. In the past, we developed Real-Time Distributed Simulation (RTDS), which is a suite of applications providing low and medium fidelity en-route human-in-the-loop (HITL) simulation capabilities contributing to the ASN. During simulation, RTDS generates and maintains two types of messages that can be shared with other simulation participants. These are messages defining the aircraft route (ETMS-route) and aircraft state (ADSB-state). RTDS is one of many simulation components that are supposed to interact within the nation-wide ASN. ASN, following the object-oriented paradigm, defines two different dedicated types of messages defining the flight (FlightObject) and aircraft (AircraftObject). To support the interconnectivity with the ASN, we designed and implemented a dedicated ASN Gateway (ASNG) acting as an intermediary, providing logic for data translation, two-way communication and transfer messages between RTDS and ASN and storage for the exchanged data. The ASNG, designed and implemented as an artifact of an earlier project in the NEAR Lab [4], supports timely translation of the messages and, thus, two-way communication between RTDS and ASN. With the gateway, different simulation systems can be shared among the aviation community, which allows the testing and evaluation of new air traffic management concepts and procedures.

The ASNG is a software system acting as an intermediary between RTDS and ASN providing:

- Logic for two-way communication and transfer messages between RTDS and ASN.
- Storage for the data exchanged between RTDS and ASN.

The original report of the ASNG [4] elaborates that the system should be secured from external interference. The reason for such operational requirements is that the RTDS uses User Datagram Protocol (UDP) sockets with different communication channels, depending on the message type (state *vs.* route). Each channel uses a port defined in a configuration file when launching RTDS. If an attacker were to connect to the network, and listen to the channel ports, he could also send messages from the gateway. In the same way, the attacker could produce misleading state or route messages that the gateway would then send to ASN.

The gateway is responsible for passing the data to-and-from and, thus, it is a vital component of the air traffic monitoring and control system. If the gateway fails to transfer correct information or a command to a flight, in the best case, the flight may be delayed; in the worst case, there is a risk of a mid-air collision. Additionally, if the gateway is not well protected, the system may be put at risk (e.g., under attack from malevolent organizations). Therefore, relevant safety/security issues must be considered during the gateway development. At its current state, the system is just a simulated real-life environment. However, the system includes also HITL components (the ERAU RTDS includes an interface with live pilots and air traffic controllers). Ultimately, the decisions regarding development of the NextGen operational software will be based on these simulations. It is thus necessary to include and analyze safety/security requirements for the gateway project based on hazards and threats related to such ultimate implementation.

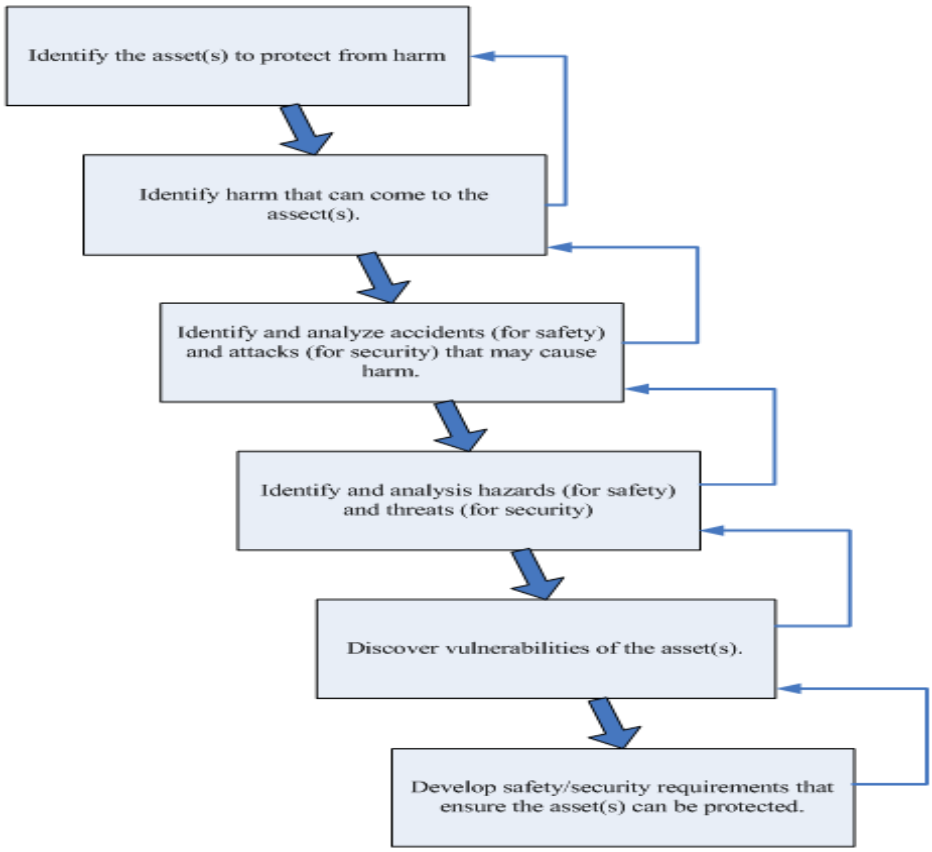
3. Safety and Security—from Hazards and Threats to Requirements

Safety or security requirements specify policies and establish goals of achieving a specified level of target safety or security. The objective is to reduce or eliminate risks due to either hazards (safety) or threats (security). In turn, hazards may result in accidents, while threats may result in attacks. Thus, effectively, safety hazards are equivalent to security threats, and the accidents associated with these hazards are equivalent to security attacks related to the threats. Either accidents or attacks may eventually cause harm to the system assets (in terms of people, property, environments or services).

Due to the above commonalities, the process of safety and security engineering can be combined, as illustrated in Figure 1. Each of the presented activities provides input to a subsequent activity, which in turn may require modification of the preceding activity, as shown by the feedback arrows.

It should be noted that safety and security issues/concerns are an integral part of the system quality, and they are not totally separate. For instance, accidents (the realm of safety) can result in security vulnerabilities that can be exploited by attacks, at which time their consequences fall within the realm of security. On the other hand, attacks may cause safety hazards that in turn may cause accidents [5].

Figure 1. Safety/security engineering process.



4. Fault Tree Analysis Technique

A wide range of techniques exist that are used to help the analyzing system from the perspective of dependability, including such properties as reliability, availability, safety, *etc.* These techniques include Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Modes and Effects Analysis (FMEA), Hazard and Operability Analysis (HAZOP), Common Cause Failure Analysis (CCFA), Operational/Support Hazard Analysis (OSHA), Cause Consequence Analysis (CCA), Bayesian Belief Network (BBN), Petri Nets (PN), Markov Chains (MKV) and more.

We have considered FMEA, which is primarily a reliability focused technique evaluating reliability and identifying single-point failures. FMEA limitations, however, include a failure to address the operational interface, multiple failures or human factors [6], which are critical in the aviation application. We also explored use of Petri Nets to identify the sequence of states the system can assume when certain events would occur. However, for the presented research, we selected the well-established Fault Tree Analysis (FTA) technique, well-supported by the available software tools.

FTA is a static analysis technique for modeling, analyzing, visually displaying and evaluating failure paths in a system [7]. FTA is a deductive analysis approach based on postulating a top level mishap and attempting to find out what modes of system, events, or component behavior contribute to this mishap [8]. FTA identifies the combination of conditions that would cause a system to reach a specific hazardous state. The fault tree is typically presented by graphic symbols, with three major building blocks: events, gates and transfer symbols. Previous research [9] proved that FTA is a powerful technique to analyze system safety/security issues and can be directly applied to software

requirement and design documents to assist validation and verification in the early phase of system development. Based on the requirement and design documents of a software project, FTA can be used to assist in evaluation and, ultimately, refine requirements and design.

When applied in the requirement phase, FTA can help find out weaknesses in the specification and identify the set of requirements that will have an impact on the system quality (in terms of safety and security). When applied in the design phase, FTA can help find out weaknesses in the high-level design and identify the components/modules and subcomponents that have a direct effect on software safety by applying the fault tree diagram to the requirements and design specification.

A variety of tools exist to support FTA, and several tools have been studied: Saphire [10], FaultTree+ [11], OpenFTA [12] and BlockSim7 [13]. The later was selected for modeling and analyses for the project due to the tool availability, flexibility, user-friendliness and good selection of analysis approaches.

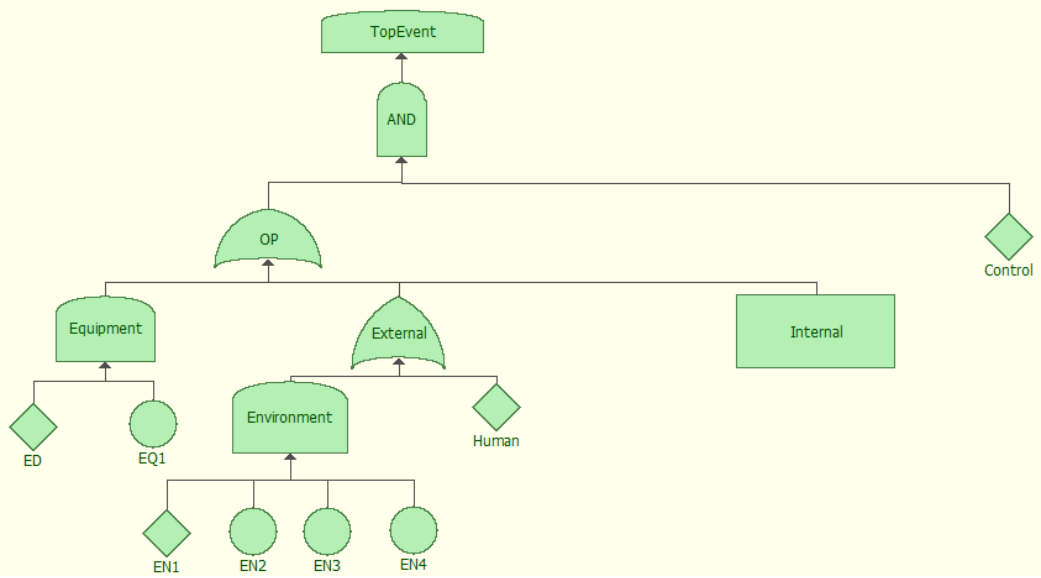
5. Fault Tree Model

The ASN Gateway is a system which acts as an intermediary between RTDS and ASN. It has two main functions: transmission of messages and storage of exchanged data. The expected features of the ASN Gateway are as follows [3]:

- Must provide the logic algorithm to translate messages between RTDS and ASN.
- Must provide storage of exchanged data between RTDS and ASN.
- Must provide two communication media allowing-
  - Transfer of message between RTD and the Gateway (ETMS-route, ADSB-state).
  - Transfer of message between ASN and the Gateway (FlightObject, AircraftObject).

The Fault Tree (FT) approach has been well-accepted to analyze system operation from the perspective of both safety and security. An appropriate FT model was developed considering the ultimate hazard being an aviation accident. The top-level FT model is shown in Figure 2.

Figure 2. Top level fault tree diagram.

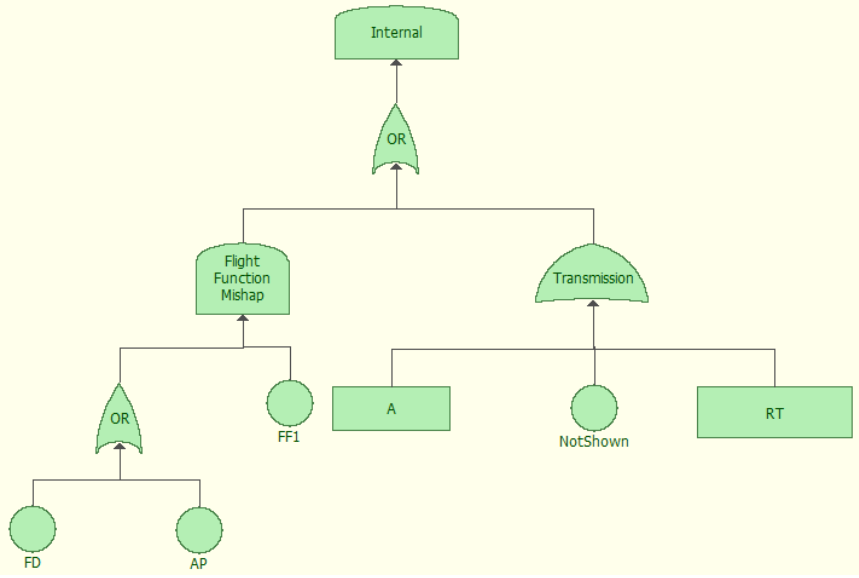


The ultimate mishap, an aviation accident, is the result of a conjunction of events during system operation and a lack of appropriate control of the operations. In turn, the failure of the system operation may result from equipment, internal and external malfunction (whereas external malfunction may be attributed to the environment or human action). The equipment factors that may cause an aviation accident include design errors, material imperfections, corrosion, *etc.* The external factors are the events outside the gateway system: environmental contributors (events that may cause damage for an aircraft transitioning through a dangerous environment) and human mistakes (made by pilots, air traffic controllers and other personnel). Both equipment and external factors are treated as undeveloped events. All these factors are outside the scope of the presented research.

The ASNG system contributes only partially to such a top-level model. We left out part of the Fault Tree that was outside the scope of the research domain, focusing on the internal factors and specifically on communication as the critical aspect in the ASNG operations. Results of further analyses include the cause-effect relationships leading to the bottom-level undeveloped events in the model [14].

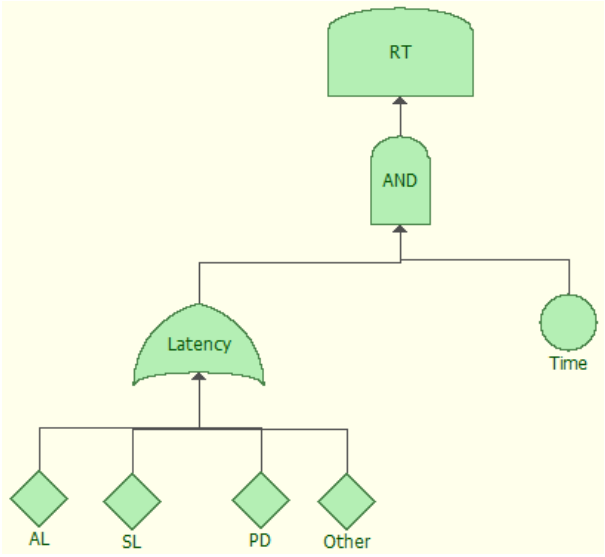
We concentrate on the internal factors (Figure 3), the main interest of the research. We assume that the systems installed on the aircraft provide two major functions: Auto Pilot (AP) and Flight Director (FD). AP is used to guide an aircraft without direct assistance from the pilot. FD is used to compute and provide flight information, so that the pilot can operate the aircraft following the FD guidance. When either the AP or FD system experience a failure, with the failure not mitigated in time (FF1), a flight function mishap may occur. However, the flight function mishap is outside the scope of the presented research. We focus on transmission of data as the critical factor in the presented research. Data transmission is the major part of fault tree models in the project. The messages are transferred between RTDS and the Gateway via local intranet UDP and between ASN and the Gateway by ASN Run-Time Interface (RTI) software using the Internet. Transmission failures can be attributed to the failure to correctly transmit (A), failure to display and, thus, not showing data to the user (NotShown) or not performing transmission in a timely manner (RT) (see Figure 3).

Figure 3. Internal component first level fault tree diagram.



Missing a deadline may cause the failure of the system, and thus, the real-time aspects must be also considered. Elements involved in the transmission of data may cause delays and system hazards due to missed deadlines. The RT sub-tree (Figure 4) would result in a failure event if the delay is too long for the transmission to meet its deadline (Time) and the latency occurs as an alternative of latencies due to application (AL), serialization (SL), propagation delay (PD) or other reasons.

Figure 4. Real-time factors, second level fault tree diagram.



The FT model for the event A representing a failure to correctly transmit a message is shown in Figure 5. Failure to correctly transmit a message is an alternative of three events: B1—failure to transfer a message from ASN to RTDS, B2—failure to transfer a message from RTDS to ASN, and B3—failure of the communication link. B3 is developed into basic events, as shown in Figure 5. The events B1 and B2 sub-trees are represented in more detailed FT models in Figure 6 and Figure 7, resulting in 37 basic events and 10 undeveloped events.

Figure 5. Data transmission, second level fault tree diagram.

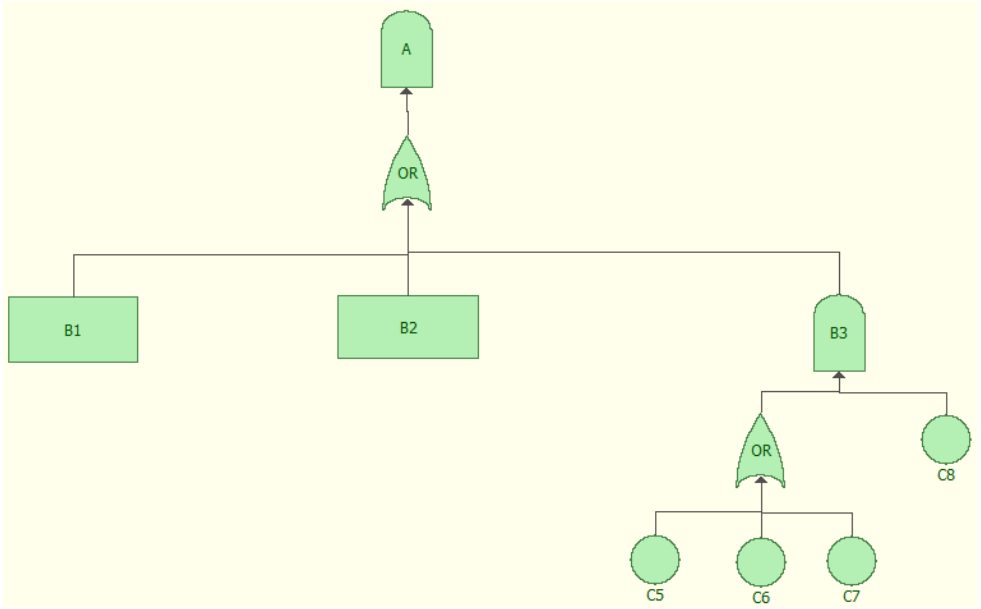


Figure 6. Aviation Simulation Network (ASN) to Real-Time Distributed Simulation (RTDS) communication, third level fault tree diagram.

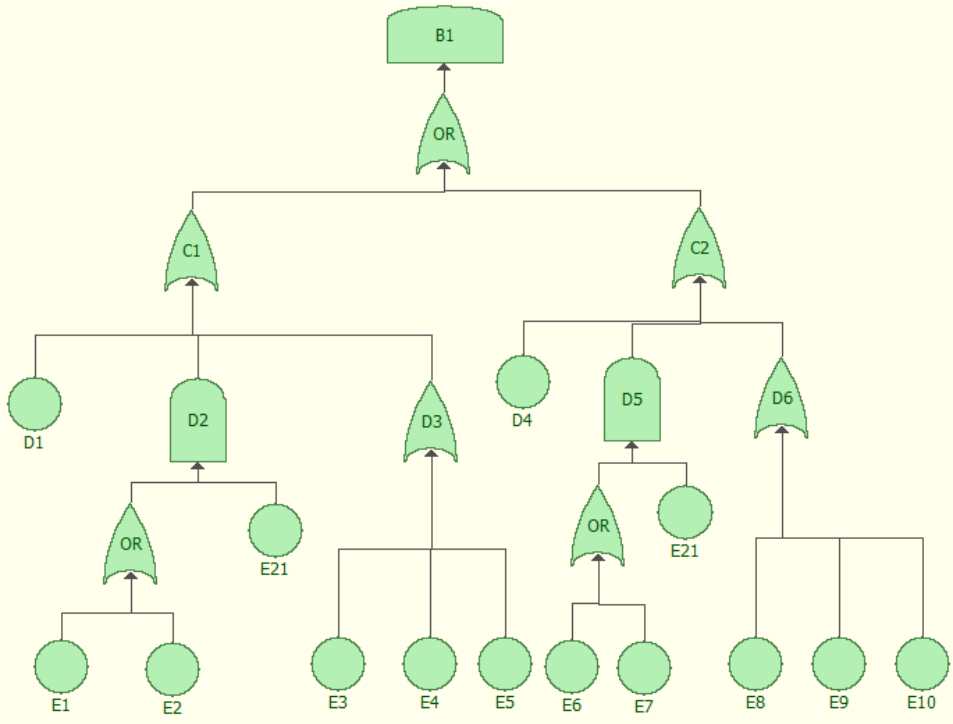
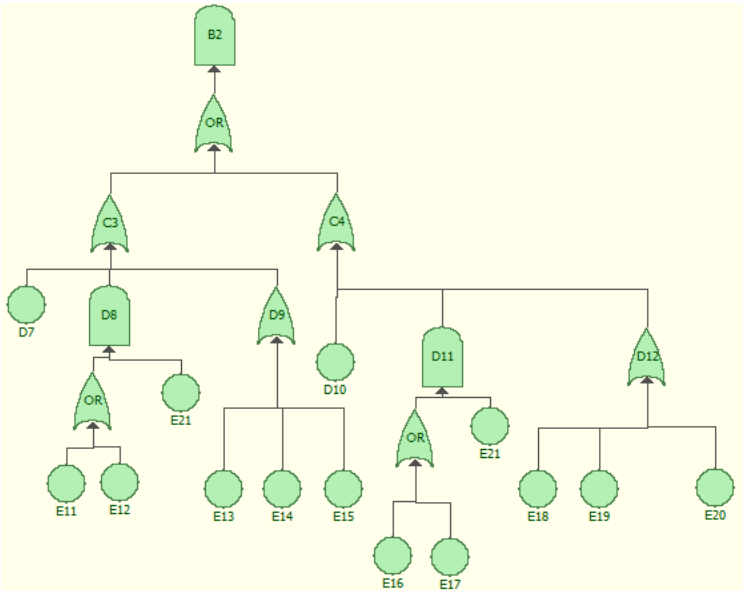


Figure 7. RTDS to ASN communication, third level fault tree diagram.



6. Simulations and Analyses

To run quantitative analyses, we need to determine the probabilities of the events. The "probability to fail" represents the probability that the event occurs within one day of operation. Based on interviews with the system users and available information on Internet component reliability, assumptions were made on the events numerical values (Table 1). These probabilities, depending on the nature of the event, are between 10^{-6} and 10^{-2} [14].

Table 1. Description and probabilities of events in the Fault Tree (FT) model.

Event ID	Description	Probability to Fail
Time	The delay too long for the transmission to meet the deadline	10^{-3}
EQ1	The equipment defects are not found and fixed	10^{-3}
EN2	The aircraft is in a dangerous environment	10^{-4}
EN3	The aircraft is influenced by environmental events	10^{-4}
EN4	Environmental events that may cause damage to the aircraft	10^{-4}
C5	Internet disconnection/downtime	3×10^{-3}
C6	Hardware (between RTDS and Gateway) failure	10^{-4}
C7	RTI software failure	10^{-4}
C8	The issues are not handled properly in time	10^{-3}
D1	RTDS fails to send ETMS-route messages	10^{-5}
D4	RTDS fails to send ADSB-state messages	10^{-5}
D7	ASN fails to send Flight Objects	10^{-6}
D10	ASN fails to send Aircraft Objects	10^{-6}
E1	Lack of encrypt mechanisms	10^{-5}
E2	Lack of authorized access/log-in mechanisms	10^{-5}
E3	Loss/damage of data storage	10^{-5}
E4	Fail to search flight/aircraft objects in database	10^{-5}
E5	Fail to update/create flight/aircraft objects in database	10^{-5}
E6	Lack of encryption mechanisms	10^{-5}
E7	Lack of authorized access/log-in mechanisms	10^{-5}
E8	Loss/damage of data storage	10^{-5}
E9	Fail to search flight/aircraft objects in database	10^{-5}
E10	Fail to update/create flight/aircraft objects in database	10^{-5}
E11	Lack of encryption mechanisms	10^{-6}
E12	Lack of authorized access/log-in mechanisms	10^{-6}
E13	Loss/damage of data storage	10^{-5}
E14	Fail to search flight/aircraft objects in database	10^{-5}
E15	Fail to update/create flight/aircraft objects in database	10^{-5}
E16	Lack of encryption mechanisms	10^{-6}
E17	Lack of authorized access/log-in mechanisms	10^{-6}
E18	Loss/damage of data storage	10^{-5}
E19	Fail to search flight/aircraft objects in database	10^{-5}
E20	Fail to update/create flight/aircraft objects in database	10^{-5}
E21	There are attackers trying to attack the system	10^{-3}
AP	Errors in AP component	10^{-4}
FD	Errors in FD component	10^{-4}
NotShown	Fail to show data to pilot/co-pilot	10^{-4}
Control	Fail to control all mishaps	10^{-3}
Equipment	Equipment errors within the aircraft(s)	10^{-4}
Environment	Environmental contributors	10^{-5}
Human	Human mistakes	10^{-4}
AL	Application layer latencies	10^{-4}
SL	Serialization latencies	10^{-4}
PD	Propagation delay	10^{-4}

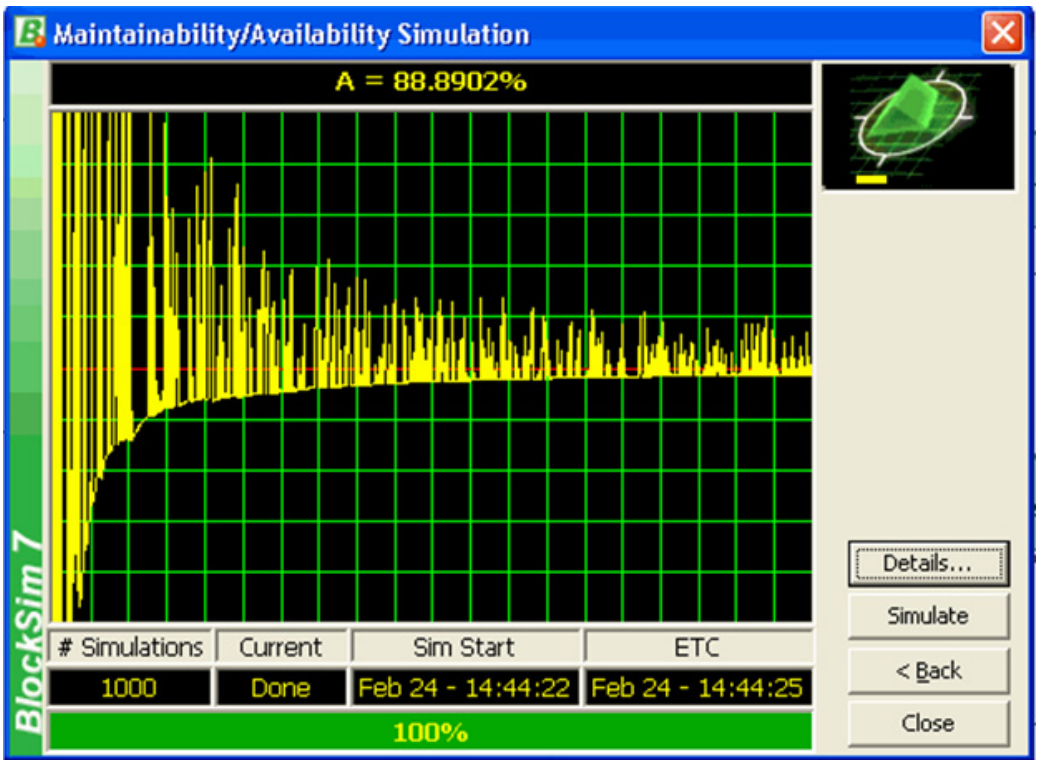
Table 1. *Cont.*

Event ID	Description	Probability to Fail
Other	All other causes	10^{-4}
ED	Equipment defects	10^{-3}
EN1	Environmental events that cause harm (lightning, storm, *etc.*)	10^{-2}

The selected FTA tool calculates non-repairable mean time to failure (MTTF) using numerical values of basic and undeveloped events. The computed MTTF is 906.5124 days, *i.e.*, the first failure of the system occurs after 2.5 years of operations.

With the number of simulations set to 1000 and the single simulation end time of 0.5 MTTF (about 453 days), the simulation results return such metrics as mean availability due to specific event occurrence, expected number of failures caused by the event, system down count caused by the event, up- and down-time as related to event and individual event failure criticality index. The results show that mean availability due to event occurrence varied from 0.87 to 0.91; and out of about 450 days of operation, the downtime was about 43–51 h (about 10%). The availability of the entire system was 88.89% (see Figure 8).

Figure 8. Availability simulation.



The unreliability curve shows how the unreliability of the system increases with time (see Figure 9). For instance, at time point 0.000 (the initial state of the system, the time units are days), the unreliability is 0, which means the system is 100% reliable. However, the unreliability increases with time; it becomes about 0.381 after about 612 days' operations (a red dot in Figure 9). And after about 3000 days' operations, the unreliability becomes 1, which means the system is 100% unreliable. The reliability decay results can be interpreted to mean that the system can maintain reliability of 95%

within a half of a year of continuous operation (~170 days). The system still maintains 99.9% reliability after one month of continuous operation (~38 days).

Figure 9. Unreliability as a function of time.

The tool allows determining the failure criticality index (FCI) for each basic/undeveloped event in the FT model. FCI is a relative index showing the percentage of times that a specific event caused a system failure. FCI can help to find out which event(s) may have the largest impact on the system and, thus, to focus appropriate mitigation methods to prevent their occurrence.

7. Safety and Security Requirements

The air traffic management assets include people and the computing system itself. The assets are related—the safety of people is dependent on the safety of the system, which in turn includes safety of data, software and hardware components. The focus of this project is the ASN Gateway software and the data that are handled by the ASNG. The system also includes a human-in-the-loop (HITL) component, with pilots and controllers that imitate a real-world environment. The safety of humans depends on correct data and message transmission. Since this is the case, we focus on the safety of data and the software, starting with identification and analysis of harm, accidents and hazard.

Safety and security both define the degree to which a mishap is prevented, which means if the hazard or threat exists; it is promptly detected and reacted to. The difference between safety and security is the source or reason for potential problems. Safety considers hazards, *i.e.*, how the system may harm the environment due to system failure or a combination of accidental conditions. Security considers threats and focuses on how the environment, e.g., potential malicious attacks, may impact

the system assets and its operation due to system vulnerability. Often times, violation of security may result in a safety violation. Safety and security requirements specify a minimum, mandatory amount of safety and security, respectively.

7.1. Safety Requirements

Considering the functions and features of the ASN Gateway, the harm to safety includes loss of data and software error/exception. Loss of data may be caused by a human mistake, catastrophic device failure or failure to do backups. Software error and/or exception can be caused by incorrect requirements, design, implementation or inadequate verification. An analysis of the above relationships allowed us to identify a set of specific representative safety requirements focusing on potential hazards related to loss of data and errors (Table 2).

Table 2. ASN Gateway safety requirements.

SA-01	The system shall have a mechanism that monitors the state of the system and warns the user/operator when error/exception occurs	
SA-02	The system shall do periodic backups and/or allow user to do backups manually	
SA-03	The system shall have recovery methods to retrieve the past data if there is damage to the database due to catastrophic failure	
SA-04	The system shall include exception-handling mechanism	
	SA-04-01	The system shall warn the user/operator about exception (See SA-01) The warning shall last for a to be determined period of time (e.g., 30 seconds) for the user/operator to respond
	SA-04-02	The system shall include methods to notify other operators within the system (e.g., staff in RTDS and ASN, the pilot, *etc.*), particularly when the user/operator of ASN Gateway fails to respond within the to be determined period of time

7.2. Security Requirements

Security is about protecting the assets from a threat. In the contemporary world, with the proliferation of interconnected computing systems, cyber-security becomes the leading point of interest. Harm to security includes communication failure, incorrect message transmissions or unauthorized use of the system (the latter often being the cause of the former ones). Since security is focusing on threats, communication failure can be caused by flooding the network or intercepting messages (man in the middle). Incorrect message transmissions can be caused by changing the content of the intercepted message. Both communication failure and incorrect transmission are due to a lack of or marginal encryption mechanisms. Unauthorized use can be caused by unauthorized access due to missing or trivial user authorization mechanisms. Similar to safety, an analysis of the above relationships allowed us to identify a set of specific representative security requirements focusing on potential hazards related to potential threats impacting communication. Respective security requirements are presented in Table 3.

Table 3. ASN Gateway security requirements.

SE-01	The system shall have encrypt/decrypt mechanisms for message transmission	
	SE-01-01	The system shall encrypt sensitive information while transferring ETMS-route message between Gateway and RTDS
	SE-01-02	The system shall encrypt sensitive information while transferring ADSB-state message between Gateway and RTDS
	SE-01-03	The system shall encrypt sensitive information while transferring Aircraft objects between Gateway and ASN
	SE-01-04	The system shall encrypt sensitive information while transferring Flight objects between Gateway and ASN
	SE-01-05	The encrypted information shall be decrypted upon reception
SE-02	The system shall encrypt sensitive data in database	
SE-03	The system shall validate that the user has the authorization to log in and change the files	
SE-04	The system shall detect consecutive failed login-in attempts	
SE-05	The system shall detect attack to the communication link	

Based on the FT models, analysis and simulations, some mitigation methods to improve the safety/security of the system have been identified. The critical fault tree model events are: Control, Human, Environment, Equipment and Transmission. The first four areas require mitigations outside the scope of software, including such elements as better training, weather forecast and equipment quality. From the perspective of the gateway software, the object of interest in our research, the specific mitigations, based on the FTA model simulation results, include:

- A mechanism that monitors the state of the system and warns the user/operator when error/exception occurs.
- An encrypt/decrypt mechanism for message transmission.
- Encryption of sensitive data in the database.
- A mechanism for detection of an attack to the communication link.

The set of mitigations proposed for Transmission events, specific for software development, have been proposed to be implemented in the subsequent revision of the system.

8. Conclusion

In the early phases of software development, high-quality requirement (and design) documents contribute to ensuring that the ultimate software system is safe and secure. It is critical to define safety and security requirements following rigorous system-level hazard and threat analyses. The foundational, measurable and repeatable scientific elements applicable to assuring the cyber-security of software systems include threat, vulnerability, security violation, attack integrity confidentiality, *etc.*

Fault Tree has been selected as the technique used for validation in the early phases of software development to analyze the system and, thus, to support development of necessary missing safety and security requirements, thus enhancing the existing software requirements specification and software design documents of the ASN Gateway project. In addition to background research, engineering knowledge and common sense, the information from an online survey and private communication

from the ASN Community staff was used, and assumptions were made on numerical values on the frequency and reliability of the events.

BlockSim7 has been selected as the tool. Subsequently, the results of the model simulations and analysis were used to propose some mitigation methods. The majority of events, like those related to human, environment and equipment, are outside the scope of the software product. However, the result related to data transmission and storage are useful for introducing mitigations designed to improve the safety and security of the system.

An additional analysis of the system safety and security was attempted using the Petri Nets (PN) approach. Preliminary results show that PN may be showing well the dynamic relations in potential failure scenarios. However, since the Stochastic PN paradigm was not used, the failure probabilities could not be assessed. For the future, the FT models may be modified according to the more detailed and updated information of the system. Also, the specific information of each element that allows us to determine the reliability will be more accurate, based on more responses solicited from experts involved in the system. For the ASN Gateway project, some other techniques (e.g., Stochastic Petri Nets, Timed Automata, Finite State Machines, Fuzzy Sets, Rough Sets, Bayesian Belief Networks, *etc*.) may be applied to its artifacts and/or products.

References

1. Tassey, G. *The Economic Impacts of Inadequate Infrastructure for Software Testing*; Technical Report for RTI Project (Number 7007.011): Gaithersburg, MD, USA, May 2002.
2. Next Generation Implementation Plan, 2011. Federal Aviation Administration Web site. Available online: http://www.faa.gov/nextgen/media/ng2011_implementation_plan.pdf (accessed on 16 April 2012).
3. What is SESAR. Single European Sky Air Traffic Management Research (SESAR) Web site. Available online: http://www.eurocontrol.int/sesar/public/standard_page/overview.html (accessed on 12 March 2012).
4. Perret, M. *HLA Gateway Between RTDS and AviationSimNet*, *SRS_ASN Gateway*, version 0.6; *SDD_ASN Gateway*, version 0.7; Graduate Research Project, Embry Riddle Aeronautical University, Daytona Beach, FL, USA, 2011.
5. Firesmith, D.G. *Common Concepts Underlying Safety*, *Security*, *and Survivability Engineering*; Carnegie Mellon Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA, USA, December 2003.
6. Stephans, R.A. *System Safety for the 21st Century*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004.
7. Ericson, C.A., II. Fault Tree Analysis—A History. In *Proceedings of the 17th International System Safety Conference*, Orlando, FL, USA, 1999.
8. Vesely, W.E.; Goldberg, F.F.; Roberts, N.H.; Haasl, D.F. *Fault Tree Handbook*; NUREG-0492; U.S. Nuclear Regulatory Commission: Washington, DC, USA, 1981.
9. Towhidnejad, M.; Wallace, D.R.; Gallo, A.M. Fault Tree Analysis for Software Design. In *Proceedings of the 27th the Annual NASA Goddard/IEEE Software Engineering Workshop*, Greenbelt, MD, USA, 2002.

10. Smith, C. SAPHIRE Risk and Reliability Assessment Software. Idaho National Engineering and Environmental Laboratory Web site. Available online: https://saphire.inl.gov/pdf/SAPHIRE_overview.pdf (accessed on 8 November 2011).
11. *Reliability Workbench Technical Specification*, version 10.3. Isograph Reliability Software Web site. Available online: http://www.isograph-software.com/_techspecs/rwb103techspec.pdf (accessed on 25 January 2013).
12. *OpenFTA*, version 1.0; OpenFTA User Manual. Formal Software Construction Limited Web site. Available online: http://www.openfta.com/ (accessed on 7 November 2011).
13. *BlockSim*, version 7; BlockSim7 Online Help. ReliaSoft Corporation Web site. Available online: http://www.reliasoft.com/BlockSim/ (accessed on 5 March 2012).
14. Liu, M. Verification and Validation in Early Phases of Software Development. Graduate Research Project, Embry Riddle Aeronautical University, Daytona Beach, FL, USA, 2012.