INCOSE

CHAPTER CONFERENCE

FMEA Representation in MBSE for Aerospace Systems

A Python-Based Extraction Approach

INCOSE MBSE SUMMIT · 2025

AUTHORS

AVRK Koundinya, S Upendra Kumar, Pavan Kumar N, Shilpa Agarwal, and Shikha Chittora

CHAPTER

India Chapter

PUBLICATION DATE

July, 2026

SUBMISSION #

25

INCOSE PUBLICATIONS

hosted by INCOSE INDIA

MBSE SUMMIT 2025

"Conquering Complexity with Simplicity: The MBSE Advantage!!"

14th - 15th June, 2025 | Pune, India

FMEA REPRESENTATION IN MBSE FOR AEROSPACE SYSTEMS: A PYTHON BASED EXTRACTION APPROACH

AVRK Koundinya
Infosys Limited
+91-7538883144
avrkkoundinya2000@gmail.com

S Upendra Kumar
Infosys Limited
+91-9966027150
upendrakumaru306@gmail.com

Pavan Kumar N
Infosys Limited
+91-7416333239
pavanknuthikattu@gmail.com

Shilpa Agarwal
Infosys Limited
Shilpa.ag2002@gmail.com

Shikha Chittora
Infosys Limited
+91-9158451836
shikhachittora2010@gmail.com

Abstract. Ensuring safety and reliability in civil aviation systems requires comprehensive analyses such as Failure Modes and Effects Analysis (FMEA). Traditional FMEA methods, often performed in isolation, introduce uncertainties in system functions and architecture. Model-Based Systems Engineering (MBSE) offers a transformative approach by establishing models as the central source of system information throughout the lifecycle, enhancing traceability, stakeholder collaboration, and complexity management. The Architecture Analysis and Design Integrated Approach (ARCADIA), implemented via the Capella tool, provides a structured framework for representing system requirements, developing system architecture, and design early in development. This approach promotes collaboration, iterative refinement, reducing late-stage design risks and improving system quality. This paper introduces an approach for representing Design Failure Modes and Effects Analysis (DFMEA) in the form of models and extracting the information from models into traditional template using python scripting. By leveraging key diagrams—including Physical Component Breakdown, Physical Data Flow Blank, Physical Architecture Blank, and Mode and State Machine Diagrams—critical model information is extracted through the capellambse Python library and transformed into a structured DFMEA report in Excel format. The integration of Capella with Python scripting enhances efficiency, accuracy, traceability and repeatability in safety analysis, significantly reducing manual effort and enabling early identification of design risks. By combining MBSE practices with automation, the proposed approach advances the rigor and efficiency of system safety assessments.

Keywords. DFMEA, FMEA, MBSE, Aerospace Systems, ARCADIA, Capella, Safety and Reliability, Python, capellambse

Introduction

In the civil aviation industry, safety of the passengers is the topmost priority. To ensure the safety of the aircraft, many tests and analyses are performed. Failure Mode and Effects Analysis (FMEA) is a systematic approach to identifying, understanding, documenting and prioritizing potential failure modes, their effects & causes. FMEA plays a very important role in identification of different modes in which each component can fail, possible events which cause the failures at the Local, System and Product level [1]. Traditionally FMEA is performed by reliability experts by using their technical expertise in the field, the process involves manually filling the sheets according to the standards specified [2]. Identification of the possible modes of failure of the components at the design stage and providing solutions for elimination of effects can be achieved by performing Design Failure Modes and Effects Analysis (DFMEA) [18]

Design and development of complex systems like aircraft engines involve design, integration, verification and validation of systems. To ensure safe operation of the complex systems the requirements from product level to component level must be defined clearly, systems engineering principles ensure that the functions of each of the components and systems as whole are captured [3]. Model Based Systems Engineering (MBSE) is increasingly adopted across industries to capture the capabilities, functions and behaviors of the complex system of systems. In MBSE, various combinations of tools, methods and languages are used to model complex engineering systems [4].

The collaboration of the International Council On Systems Engineering (INCOSE) and Object Management Group (OMG) for developing and promoting MBSE standards and practices resulted in the development of Systems Modelling Language (SysML). SysML is developed based on the Unified modelling language (UML) which allows user to specify, design, analyze and verify complex systems [5].

Architecture Analysis and Design Integrated Approach (ARCADIA) is a methodology which is developed by Thales group implemented in the open-source tool Capella, which is gaining popularity in the various fields including aerospace [6]. Capella is a hybrid modelling tool [7] which provides an environment for implementing the ARCADIA method and uses a language which is heavily based on SysML [7].

Significant number of studies have been performed on developing methods for generation of FMEA (Failure mode and effect analysis) using SysML diagrams such as block definition diagrams, internal block diagrams, state transition machines and activity diagrams [8] in the software and cyber security domain. For processing the diagrams and extracting information required for generating FMEA, AltaRica modelling and language tool is used in the study performed by Hecht M et al [9]. Generating Fault tree analysis and FMEA artifacts is explored by Gaëlle Girard et al. [15]. A study was conducted by Winton, D. B rt al. [16] for generating FMEA using a specialized tool Vitech GENESYS which explored including artifacts at architecture level of model.

Complex mechanical and aerospace system requires deeper understanding to capture the failure modes of design and identify the impact of it at component and system level. Generation of FMECA for NASA RS-25 rocket engine by using SysML is carried out by Shreyas Raghu et al. [10] for performing the cost, reliability and safety studies. This paper integrates safety methods in MBSE for rocket engine development. Model based approach is used for generating FMECA in smart manufacturing applications by Ghoshal, S et al. [17] which explains FMECA representation using SysML.

Building upon these efforts, this paper presents building MBSE model for FMEA and a novel method for automated FMEA details extraction by integrating Capella models with Python scripts utilizing the capellambse library. This library enables direct access to model attributes and relationships, facilitating efficient extraction of the necessary data for FMEA generation. This method models the stakeholder requirements which are used to capture functions and failure modes of the system in ARCADIA. The methodology has been implemented on aircraft fan sub-system as an example using different diagrams created in Capella. The structure and modelling approach along with integration of python with Capella for generating DFMEA is detailed in the following sections of this paper.

ARCADIA Methodology and Capella

ARCADIA is a methodology which supports Model Based Systems Engineering used to develop systems & architecture engineering. The modelling language is inspired by SysML and supported by the tool Capella. Diagrams, viewpoints, and architecture of the system are modelled to understand functions and capabilities of the system.

The methodology defines four layers of architecture [11] [13]

1. Operational architecture level - For understanding the real customer need and defining capabilities and mission of the system.
2. system architecture level - Defining the functions to be performed by the system and exchange of data between the functions.
3. Logical architecture level - Identifying the logical functions of the system and developing architecture which is component solution neutral at the logical architecture level.
4. Physical architecture level - Allocating the functions to be performed by the system to different components which are designed to deliver the functions.

Role of different layers in ARCADIA is illustrated in Figure 1.

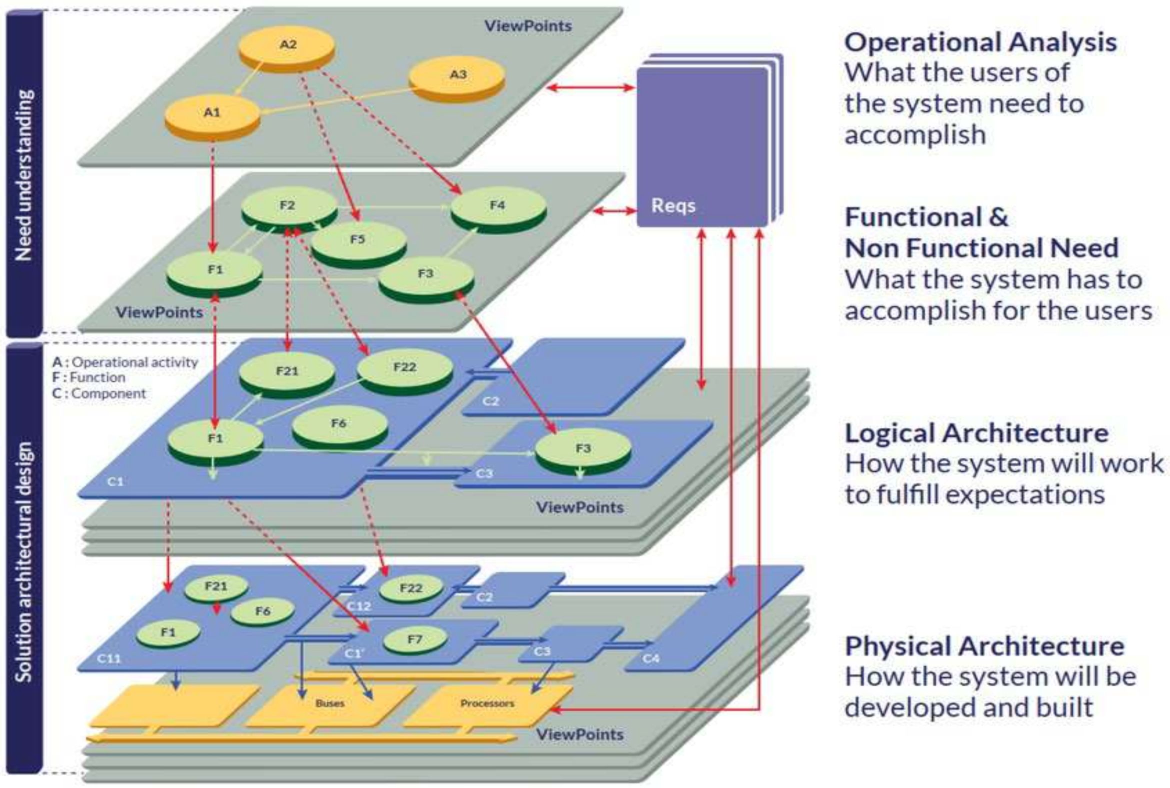


Figure1. Different layers of ARCADIA and their significance [13]

The process level advantages achieved by ARCADIA include defining and sharing the product architecture among all engineering stakeholders, validation and justification of design at early stage of the project and easing and mastering the process of Integration, Validation, Verification, Qualification (IVVQ) [14]. The process of defining and analyzing needs of the customer at different levels in ARCADIA method is illustrated in Figure 2.

The ARCADIA method is helpful in applications of any type of interdisciplinary complex system of systems which include hardware, software, mechanical, electrical and electronic systems. Multiple viewpoints can be created so that different perspectives like cost, safety, weight can be taken into consideration to make trade-off studies [14]. The usage of this method can be extended to different levels of analysis from product level to component and sub-component level and involves a range of stakeholders.

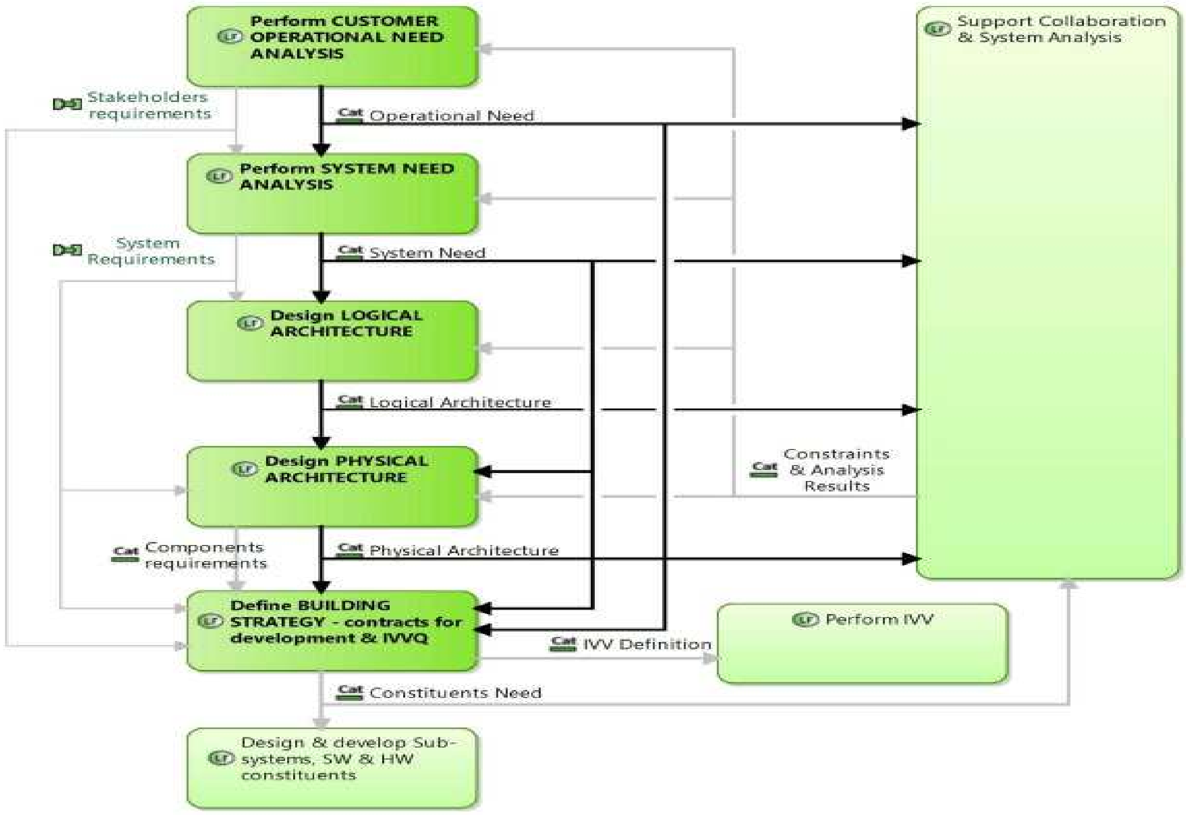


Figure 2. Analysis flow at different levels in ARCADIA [14]

System Representation for DFMEA in MBSE

For representation of DFMEA the required information to be extracted from the model includes the complete list of components and subcomponents in the system, failure modes of the system and events causing the failure. Effects of the failure modes on the system at different levels are to be captured as Local effect, system effect, product level effects. Mitigation or steps which are to be followed to prevent failure, severity, occurrence, detection of the failure and Risk Priority Number (RPN) are to be captured for assessing the criticality of the failure mode [12]. Below image shows the pictorial representation of FMEA process. [19]

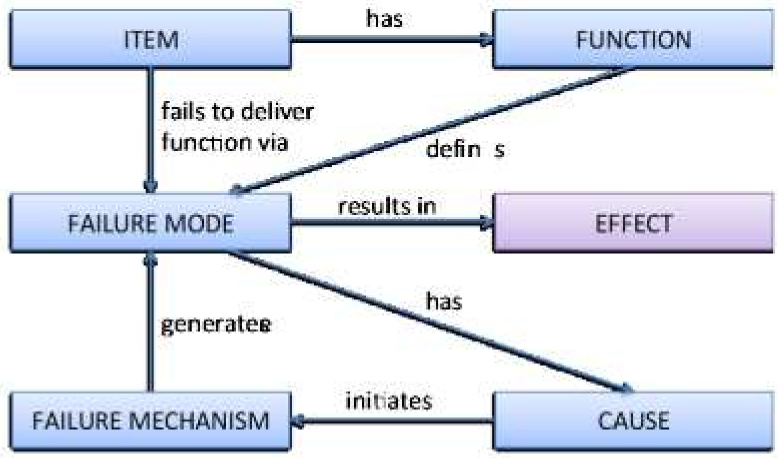


Figure 3: The Systems Model of Failure for Product-Based Systems (Design FMEA)

The following diagrams from Capella are used for extracting each of the above-mentioned data for extracting DFMEA:

Components and Breakdown

DFMEA requires a definitive list of components which the system constitutes. This list gives information about what are the functions that components fail to deliver. In Capella the components of the systems are defined as Nodes at physical level. The nomenclature and hierarchy of the components are defined in the Physical Component Breakdown Diagram (PCBD) at the physical analysis level.

Failure Modes and Causes

Different ways in which failure of the component is possible are captured as the failure modes in the Mode and State Machine diagram [11] of Capella. For each component, separate Mode and State Machine diagrams are created. Failure modes are captured in the mode blocks and transition from the safe mode to the failure mode is captured in the transitions. The cause of the failure is captured in the triggers section of the transitions.

Local, System and Product Level Effects

Effects of the failure modes are captured at the local level (on the component), system level which is at the system/ sub-system of which the component is part of, and product level which considers the change in performance of the product due to the failure of the component. In the present approach the effects are captured as the property values of data type “string” which are groups allocated to the modes in which respective failure modes of the components are captured. The propagation of the failure modes at the system level are captured as functional exchanges between the functions allocated to the components in the Physical Architecture Blank (PAB).

Severity, Occurrence and Detection

Severity is the measure of the impact of failure on the system and the product in the event of failure, Occurrence is the probability or likelihood of a failure occurring during the life of the product, which based on number of failures that are observed from the service data and Detection is the ability to identify the failure if the component fails. These are calculated on a scale of 10 and the product of these parameters is calculated to represent the Risk Priority Number (RPN). Each of these parameters is captured as Property value group which is allocated to the failure mode in the Mode and State Machine diagram.

Mitigation

The actions which are recommended to address the failure modes, such as changes in the design or improvements in the methods of detection, are captured under mitigation in the detailed design stage of the product. In this paper these are captured in the model as a property value which is named as Mitigation and allocated to the Failure mode to which it relates in the Mode and State Machine diagram.

Aircraft Engine Fan Sub System Example

The Modelling of DFMEA is explained with the help of an Aircraft Engine Fan sub-system example. In aircraft engines, Fan or Low-Pressure Compressor (LPC) is at the intake of the engine whose main functionality is to compress the air to a defined pressure and split the flow between core and bypass to generate thrust in the turbofan engine. Most of the thrust in the engine is generated by the bypass flow. Modelling details on how to capture components, functions, exchanges, property value groups and Mode and State transitions are discussed below.

Physical Component Breakdown Diagram

The list of components which constitute the system is captured in the Physical Component Breakdown Diagram. Nose cone, Fan Blade, Fan Disc, Fan Casing and Outlet Guide Vanes are the components responsible for the generation of thrust and compression of air which is to be provided as input for high pressure or core compressor of the fan sub-system. The Component Breakdown diagram for the Fan sub-system from which the list of components is extracted for performing DFMEA is shown in Figure 4 below.

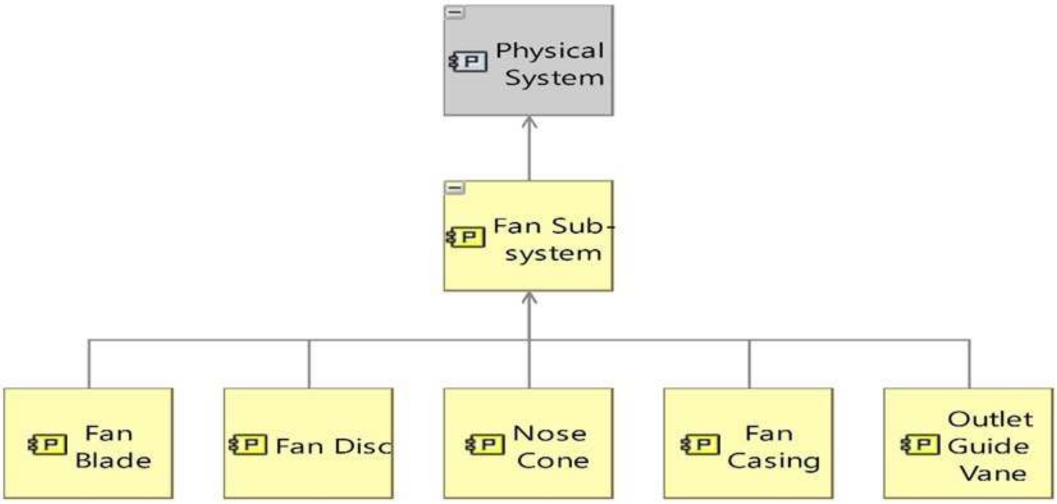


Figure 4. Physical Component Breakdown Diagram of Fan Sub-System

Physical Data Flow Blank

The functions which are to be performed by the system and the interactions between the functions are captured in the Physical data Flow Blank. In the current example the functions of components for achieving the overall functions of the system are captured. Streamlining of air, transfer of torque/ force, acting as boundary and increase kinetic energy of air flow and Transfer of heat are captured as primary functions of the System. The interactions between the functions are captured with the help of functional exchanges. The physical data flow blank for fan sub-system is shown in Figure 5 below.

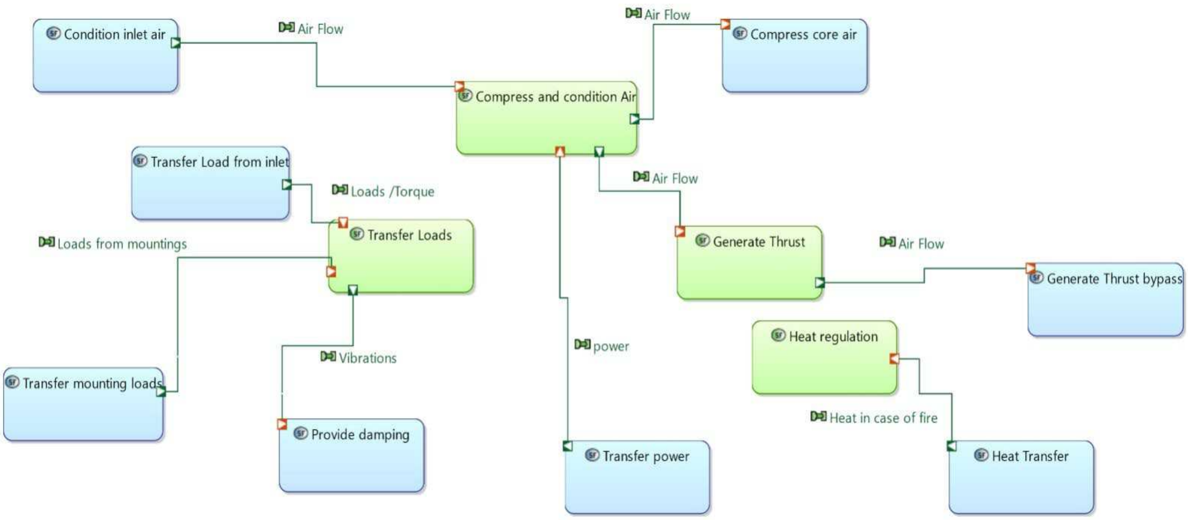


Figure 5. Physical Data Flow Diagram of Fan Sub-System

Physical Architecture Blank

The functions to be performed by the system are allocated to the respective components and the physical links between the components are captured in the physical architecture blank. The functions to be performed in the safe mode and failure preventive functions are captured in different behavior components which are allocated to the physical components. The inputs which trigger the failure mode are captured as functional exchanges from the point of origin of the cause to the component to which it effects. The Physical Architecture Blank of the Fan sub-system of Aircraft engine is shown in Figure 6 below.

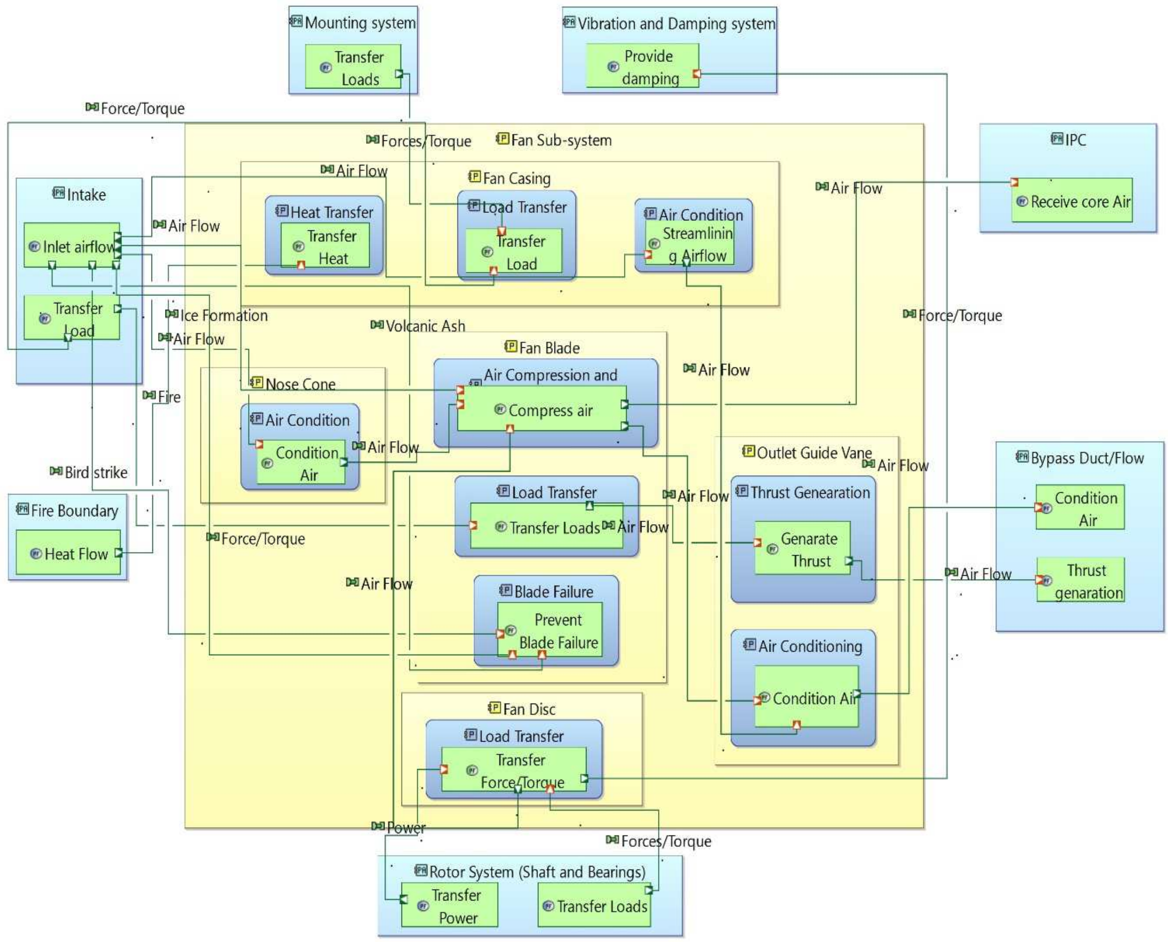


Figure 6. Physical Architecture Blank of Fan Sub-System

Mode and State Machine Diagram

The different ways in which component failure occurs are captured in the Mode and State Machine diagram at the physical architecture level. Failure modes are captured as transition from safe operation and causes are captured as triggers for transitions. Triggers are either change events or functional exchanges which are inputs for the failure modes. Local, system and engine level effects which are captured as property values are allocated to Modes. The properties which are used to determine the failure category and steps for mitigation are also allocated to respective failure modes. The Mode and State Machine diagram representing one of the failure modes of Fan Blade and Fan Disc are shown in Figure 7 and Figure 8 below.

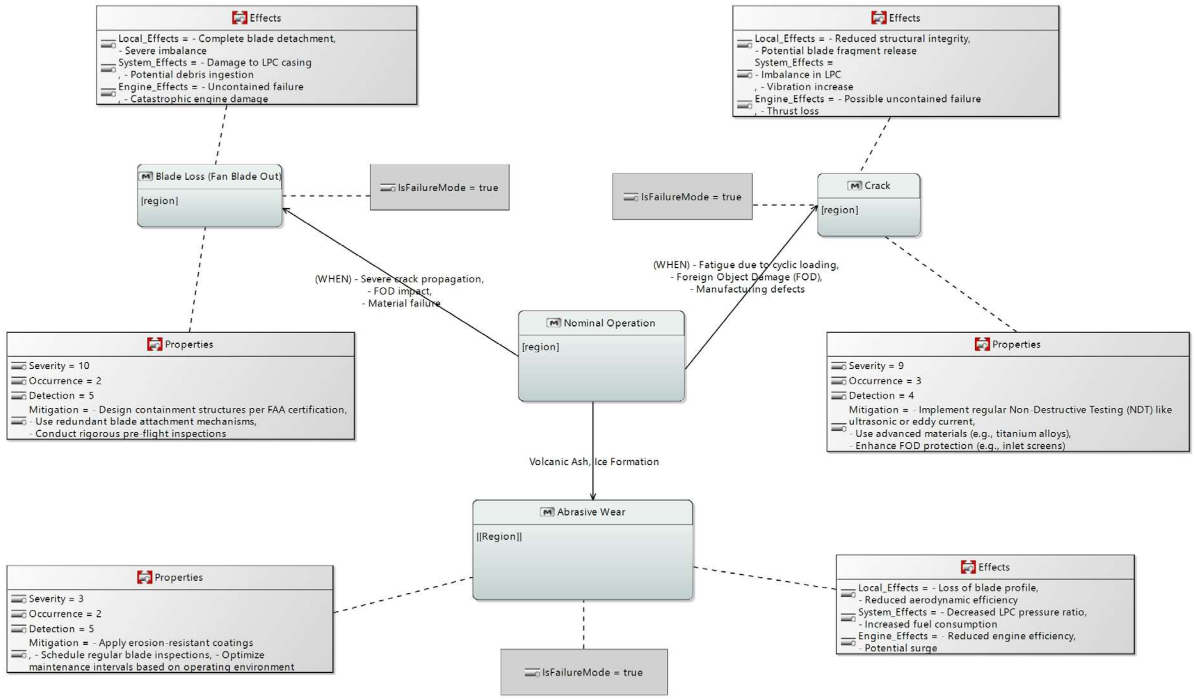


Figure 7. Mode and State Machine Diagram of Fan Blade

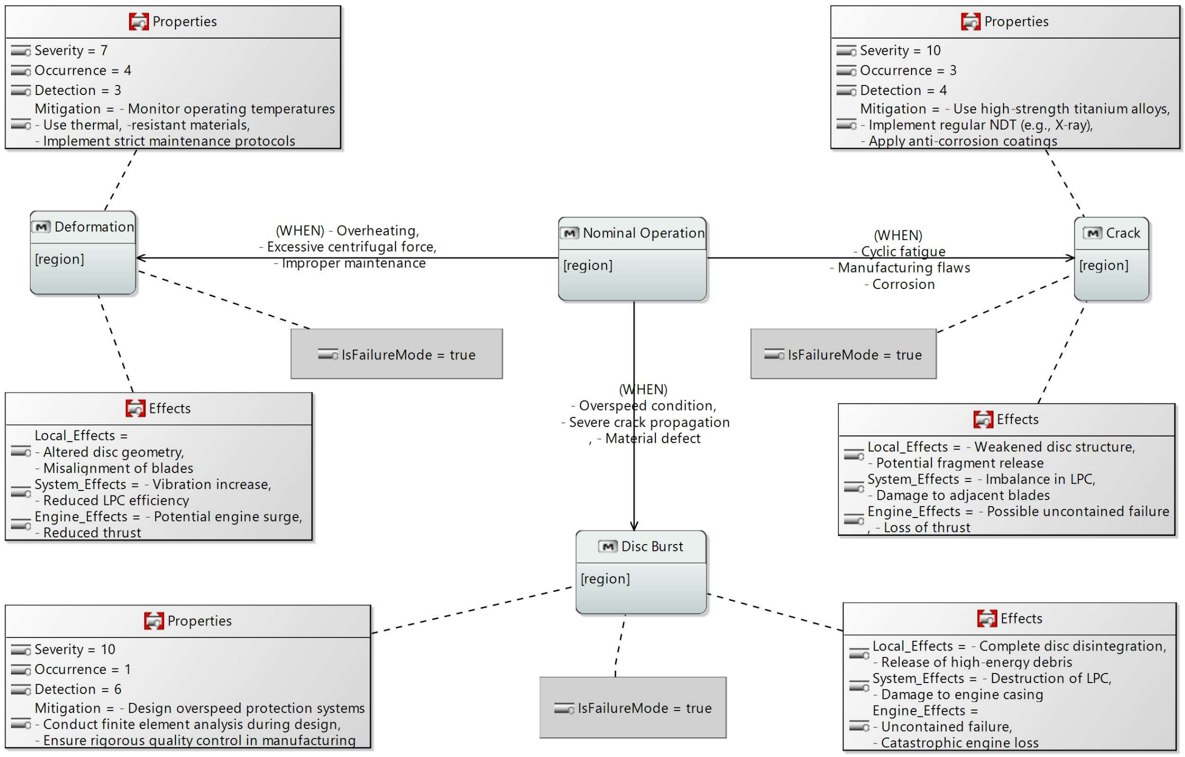


Figure 8. Mode and State Machine Diagram of Fan Disc

Extraction of DFMEA using Python

The extraction of Design Failure Modes and Effects Analysis (DFMEA) data from fan sub-system Capella model is automated using the capellambse Python library. The process transforms complex model data into a structured table format suitable for further analysis, reporting, or certification tasks. Detailed failure information (such as effects, causes, and criticality metrics) is retrieved from the Physical Architecture layer of a Capella model and exported into an Excel or CSV file without manual intervention. The script verifies the existence of the provided Capella.aird model file. It ensures compatibility by checking the capellambse library version, noting the use of capellambse 0.6.16 with Capella 6.0.0. The model is loaded into memory through capellambse.MelodyModel. To maintain robustness, error handling is incorporated for model loading and structural validations. Once loaded, the script navigates to the Physical Architecture (PA) layer. If direct access to PA fails, a fallback search is performed across model layers to identify a suitable architecture. Within the PA, all Physical Component elements are retrieved either via model.search("Physical Component") or fallback attributes like owned_physical_components. The script iteratively examines each component for associated State Machines that define system behavior under various conditions.

For each state within these machines, the script determines whether it represents a Failure Mode based on a specific property value attached to the Mode block named IsFailureMode, if it is marked as True the mode is considered as a failure mode. When a failure mode is identified, the script extracts multiple properties like Effects (Local, System, Engine) from associated Property Value Groups along with Severity, Occurrence, Detection, and Mitigation attributes. Causes are inferred from incoming transitions to the state, based on triggers or descriptions. Risk Priority Numbers (RPNs) are computed by multiplying Severity, Occurrence, and Detection scores, where valid numeric values are present. All extracted information is systematically organized into a Pandas DataFrame under the following columns: Component, Failure Mode, Local_Effects, System_Effects, Engine_Effects, Cause, Severity, Occurrence, Detection, RPN, and Mitigation. The resulting DataFrame is exported to an Excel file using openpyxl, with CSV export provided as a fallback in case of errors.

Results and Discussion

The python script reads the MBSE model and extracts the DFMEA table in excel format. The sample output for the components Fan Blade and Fan Disc is shown in Table1.

Component	Failure Mode	Cause	Local_Effects	System_Effects	Engine_Effects	Severity	Occurrence	Detection	RPN	Mitigation
Fan Blade	Abrasive Wear	Volcanic Ash, Ice Formation.	Loss of blade profile, Reduced aerodynamic efficiency.	Decreased LPC pressure ratio, Increased fuel consumption.	Reduced engine efficiency, Potential surge	3	2	5	30	Apply erosion-resistant coatings, Schedule regular blade inspections, Optimize maintenance intervals based on operating environment.
Fan Blade	Crack	Fatigue due to cyclic loading, Foreign Object Damage (FOD), Manufacturing defects.	Reduced structural integrity, Potential blade fragment release.	Imbalance in LPC, Vibration increase.	Possible uncontained failure, Thrust loss	9	3	4	108	Implement regular Non-Destructive Testing (NDT) like ultrasonic or eddy current, Use advanced materials (e.g., titanium alloys), Enhance FOD protection (e.g., inlet screens).
Fan Blade	Blade Loss (Fan Blade Out)	Severe crack propagation, FOD impact, Material failure.	Complete blade detachment, Severe imbalance.	Damage to LPC casing, Potential debris ingestion.	Uncontained failure, Catastrophic engine damage	10	2	5	100	Design containment structures per FAA certification, Use redundant blade attachment mechanisms, Conduct rigorous pre-flight inspections.
Fan Disc	Crack	Cyclic fatigue Manufacturing flaws, Corrosion.	Weakened disc structure, Potential fragment release.	Imbalance in LPC, Damage to adjacent blades.	Possible uncontained failure, Loss of thrust	10	3	4	120	Use high-strength titanium alloys, Implement regular NDT (e.g., X-ray), Apply anti-corrosion coatings.
Fan Disc	Disc Burst	Overspeed condition, Severe crack propagation, Material defect.	Complete disc disintegration, Release of high-energy debris.	Destruction of LPC, Damage to engine casing.	Uncontained failure, Catastrophic engine loss	10	1	6	60	Design overspeed protection systems Conduct finite element analysis during design, Ensure rigorous quality control in manufacturing.
Fan Disc	Deformation	Overheating, Excessive centrifugal force, Improper maintenance.	ltered disc geometry, Misalignment of blades.	Vibration increase, Reduced LPC efficiency.	Potential engine surge, Reduced thrust	7	4	3	84	Monitor operating temperatures, Use thermal, -resistant materials, Implement strict maintenance protocols.

Table1. Sample DFMEA output for Fan Blade and Fan Disc

Note: The data present in Table 1 is for representational use only. The values do not represent the actual values for the failure modes.

Conclusion

This study presents a structured and automated methodology for generating Design Failure Modes and Effects Analysis (DFMEA) by leveraging ARCADIA methodology within Capella modeling environment. Through Python integration using the capellambse library, model data has been successfully extracted to automate the extraction of comprehensive DFMEA outputs in Excel format. This solution provides a method for visualization of the failure modes in form of models which increases the clarity and understanding of complex systems and establishes relationships to the system components which can be further explored for capturing the propagation of failures at system level.

The methodology was validated using an aircraft fan sub-system as a representative use case. The fan sub-system, critical to thrust generation and engine performance, was modeled using key Capella diagrams, including the Physical Component Breakdown Diagram (PCBD), Physical Architecture Blank (PAB), and Mode and State Machine diagrams. Failure modes such as blade cracking, disc burst, and wear were captured in detail, along with their causes, system-level effects, and mitigation strategies. The automated pipeline accurately extracted the information from the models and printed it in traditional excel template which can be validated by the experts.

By demonstrating the automation of DFMEA and integrating safety analysis into the system architecture development process within a realistic aerospace application. This shows that this approach can be used for Safety critical systems such as Fan system. This work meets the objectives of Model-Based Systems Engineering (MBSE): improving consistency, enabling early validation, and managing system complexity through a single source of truth.

This contribution provides method for visual representation of DFMEA with different diagrams in the Capella tool and extraction of data into traditional template. This also sets the stage for future extensions including real-time DFMEA updates, ontology-based model enrichment, and broader domain applicability.

Recommendations

The methodology demonstrated in this paper can be used for future developments in model-driven safety analysis in aerospace and other safety-critical domains. Future enhancements can focus on integrating real-time system simulation to dynamically update DFMEA values as design iterations evolve. This would enable closed-loop safety validation during the system lifecycle.

Further work can also involve linking Capella models with formal verification tools and risk management platforms to support compliance with standards such as ARP4761 and DO-178C. Enhancing the capellambse integration to support automated data validation and traceability from requirement to risk metrics would further strengthen certification-readiness.

This approach can be expanded to cover multi-disciplinary interactions—such as thermal, hydraulic, and control systems—can improve its applicability across the broader aircraft engine architecture. Additionally, incorporating natural language processing (NLP) to suggest failure modes based on historical databases can assist designers with limited domain knowledge.

The use case can be extended beyond the fan sub-system to model and evaluate complete propulsion systems. This would support more holistic risk assessments and further streamline safety analysis across development programs.

These enhancements will increase the role of MBSE from documentation and design capture to an active, predictive tool for engineering safe, reliable, and certifiable aerospace systems.

Acknowledgement

We would like to express our sincere gratitude to our organization, Infosys Limited, for providing us with the opportunity to explore and contribute to new and emerging areas. Your continued support and encouragement have been instrumental in fostering innovation and professional growth.

A special thanks to Mr. Guruprasad Kulkarni, Mr. Senthilkumar Narendran and Mr. Karthik Srinivasan for their invaluable guidance, motivation, and mentorship throughout this journey. Their insights and encouragement have played a pivotal role in shaping our efforts and driving us toward excellence.

References

[1] (Deshmukh Y., Mishra A. K., Mane S., Gupta R., & Kunjir V., 2024). Effect of failure mode causing in aerospace manufacturing industries and its applications: Conceptual approach. *AIP Conference Proceedings*, *3037*, 020024. https://doi.org/10.1063/5.0196289

[2] (Huang Z., Swalgen S., Davidz H., Murray J., & Aerojet Rocketdyne. (n.d.)). *MBSE-Assisted FMEA Approach – Challenges and opportunities* (journal-article).

[3] (Wenyue W., Junjie H., Yinxuan M., Jinjie N., & Zhiang L., 2022b). Application and development of MBSE in aerospace. *Journal of Physics Conference Series*, *2235*(1), 012021. https://doi.org/10.1088/1742-6596/2235/1/012021

[4] (Ramos A. L., Vasconcelos Ferreira J. V., & Barcel´O J., 2012, January). Model-Based Systems Engineering: an emerging approach for modern systems. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS* (Vol. 42, pp. 101–102).

[5] (David P., Idasiak V., Kratz F. D. R. & Institut PRISME – ENSIB., 2009, December). Reliability study of complex physical systems using SysML (journal-article). *Reliability Engineering and System Safety*. https://doi.org/10.1016/j.ress.2009.11.015

[6] (Baron C., Grenier L., Ostapenko V., & Xue R., 2023). Using the ARCADIA/Capella Systems Engineering Method and tool to design Manufacturing Systems—Case study and Industrial feedback. *Systems*, *11*(8), 429. https://doi.org/10.3390/systems11080429

[7] (Alai S. P., 2019). EVALUATING ARCADIA/CAPELLA VS. OOSEM/SYSML FOR SYSTEM ARCHITECTURE DEVELOPMENT. (Purdue University). Purdue University.

[8] (Hecht M., Dimpfl E., Pinchak J., & The Aerospace Corporation, 2015, July). Using SysML to automatically generate of failure modes and effects analyses. 25th Annual INCOSE International Symposium (IS2015).

[9] (Hecht M., Nguyen E., Chuidian A., Pinchak J., & Dimpfl E., 2015). Creation of Failure Modes and Effects Analyses from SysML. *SAE Technical Papers on CD-ROM/SAE Technical Paper Series*. https://doi.org/10.4271/2015-01-2444

[10] (Raghu S. L., & Thomas L. D., 2021). Model Based Failure Mode, Effects and Criticality Analysis (MBFMECA) for the RS-25 Engine using SysML. AIAA *SCITECH 2022 Forum*. https://doi.org/10.2514/6.2021-0305

[11] (Roques P., 2017). Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella. ISTE Press - Elsevier.

[12] (DEPARTMENT OF DEFENSE, 1980, October). *MIL-STD-1629A*.

[13] (Gerard S., Babau J., & Champeau J., 2005b). Model driven engineering for distributed Real-Time embedded systems. Wiley-ISTE.

[14] (Galisson G., Hammadi M., Gherib M., & Choley J., 2022). Model-based Systems Engineering methodology for defining multi-physics simulation models. *2022 IEEE International Systems Conference (SysCon)*, 1–8. https://doi.org/10.1109/syscon53536.2022.9773905

[15] (Gaëlle Girard, Ivan Baeriswyl, Jonathan James Hendriks, Roland Scherwey, Christian Müller, Philipp Hönig & Rüdiger Lunde, 2020). Model based Safety Analysis using SysML with Automatic Generation of FTA and FMEA Artifacts. *European Safety and Reliability Conference*. https://doi.org/10.3850/978-981-14-8593-0

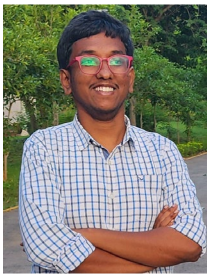
[16] (Winton D. B., & Huang Z. C., 2021). MBSE and FMEA integration using GENESYS. *2022 Annual Reliability and Maintainability Symposium (RAMS)*, 1–7. https://doi.org/10.1109/rams48097.2021.9605748

[17] (Ghoshal S., Deb S., Haste D., Hess A., Zahiri F., Sutton G., & Warner-Robins Air Logistics Complex, 2019). An integrated model-based approach for FMECA development for smart manufacturing applications. *Annual Conference of the Prognostics and Health Management Society 2019*.

[18] (Sharma K. D., & Srivastava S., 2018). Failure Mode and Effect Analysis (FMEA) Implementation: a literature review. *Journal of Advance Research in Aeronautics and Space Science*, *1–2*, 1–17.

[19] (Burge S, 2018, January). A Systems Approach to Failure Modes, Mechanisms, Effects and Causes. Systems Engineering. Technical Paper 0003/sb

Biography



AVRK Koundinya. Motivated Systems Engineer with 2 years of experience in requirements management, system analysis, and managing complex system models using Model-Based Systems Engineering (MBSE) methodologies. Adept at delivering scalable, high-quality solutions in fast-paced environments, with a strong focus on innovation in systems engineering practices. Enthusiastic about the integration of artificial intelligence into MBSE to enhance efficiency and decision-making.

S Upendra Kumar. Systems Engineering Professional with 7 years of experience with a strong specialization in aircraft fuel subsystems and component development. Demonstrated expertise in solving internal fluid mechanics challenges using Computational Fluid Dynamics (CFD). Proven track record in the verification of secondary air systems and oil systems for aircraft propulsion. Possesses a deep understanding of verification, validation, and certification processes, with a focus on designing and developing complex aerospace systems that prioritize safety, reliability, and performance.

Pavan Kumar N. Systems Engineering Professional with 5 years of experience in the Aerospace domain, specializing in requirements management and model-based systems engineering (MBSE) for aircraft engine systems. Proficient in developing customized user interfaces in IBM DOORS using DXL scripting. Experienced in conducting data analysis on intricate engine datasets utilizing Python and Power BI. Skilled in designing and configuring advanced geometries for fifth-generation fighter aircraft using CATIA.

Shilpa Agarwal. Systems Engineering Professional with 19 years of experience in aerospace sector, specializing in requirements management, V&V of control systems software, and certification audits. Expertise in ensuring compliance with industry standards and delivering high-integrity software solutions for complex aerospace systems. Holds certifications including ISO 9001 Lead Auditor, AS9100D Internal Auditor, and Certified Scrum Master, reflecting a commitment to quality, process excellence, and agile methodologies.

Shikha Chittora. Systems Engineering Professional with 12 years of experience in the Aerospace domain, specializing in requirements management, compliance management, and verification of aircraft components. Proficient in Model-Based Systems Engineering (MBSE) for engine development, with hands-on experience in control subsystem design and verification for aircraft engines. Actively contributed to AS9100 audits for control system design, demonstrating a strong commitment to quality standards and regulatory compliance.