A Safety-Focused Systems Architecting Framework for Aircraft Conceptual Design

Andrew Kingsley Jeyaraj

A Thesis
in
The Department
of
Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Mechanical Engineering) at
Concordia University
Montréal, Québec, Canada

May 2025

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Andrew Kingsley Jeyaraj**
Entitled: **A Safety-Focused Systems Architecting Framework for Aircraft Conceptual Design**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.
Signed by the Final Examining Committee:

Chair
Dr. Luiz Lopes

External Examiner
Dr. Mirko Hornung

Examiner
Dr. Charles Kiyanda

Examiner
Dr. Catharine Marsden

Examiner
Dr. Ali Akgunduz

Supervisor
Dr. Susan Liscouët-Hanke

Approved by
Muthukumaran Packirisamy, Chair
Department of Mechanical, Industrial, and Aerospace Engineering

2025

Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

A Safety-Focused Systems Architecting Framework for Aircraft Conceptual Design

Andrew Kingsley Jeyaraj, Ph.D.
Concordia University, 2025

To reduce the environmental impact of aviation, aircraft manufacturers develop novel aircraft configurations and investigate advanced systems technologies. These new technologies are complex and characterized by electrical or hybrid-electric propulsion systems. Ensuring that these complex architectures are safe is paramount to enabling the certification and entry into service of new aircraft concepts. Emerging techniques in systems architecting, such as using model-based systems engineering (MBSE), help deal with such complexity. However, MBSE techniques are currently not integrated with the overall aircraft conceptual design using automated multidisciplinary design analysis and optimization (MDAO) techniques. Current MDAO frameworks do not incorporate the various aspects of system safety assessment. The industry is increasingly interested in Model-Based Safety Assessment (MBSA) to improve the safety assessment process and give the safety engineer detailed insight into the failure characteristics of system components early in the design process. This thesis presents a comprehensive framework to introduce aspects of the SAE ARP4761 safety assessment process in conceptual design while also considering downstream compatibility of the system architecting and safety assessment processes. A generic element architecture description approach, implemented using a graph-based system architecture descriptor, is introduced to model and transfer system architecture information between each stage of the systems architecting process while supporting safety assessment activities at multiple levels of architecture granularity. The proposed framework introduces a safety-based filtering approach for large system architecture design spaces and integrates quantitative safety assessment methods compatible with early-stage system architecture specifications. Furthermore, the generic element descriptor links early system architecture specification with formal architecture specification in an MBSE environment. The framework also enables both simple and formal system architecture specification models to be used as inputs to safety assessment, as well as a source of system-level sizing parameters for MDAO workflows featuring system sizing tools. The framework's effectiveness is illustrated with examples from applications in recent collaborative research projects with industry and academia, which feature safety-focused system architecting studies for a conventional aircraft landing gear braking system, a yaw control actuation system, and unconventional yaw control system architectures for hybrid-electric and distributed-electric aircraft. The work presented in this thesis contributes to increasing maturity in conceptual design studies and fosters innovation by opening the design space while considering safety upfront.

Acknowledgments

I would like to begin by thanking God for carrying me all this way, from being a kid enthralled by airplanes and spaceflight to deciding to embark on this PhD journey. I thank him for lifting me up over all the challenges, obstacles, frustrations and for the successful publication of this thesis. I thank him for blessing me with excellent colleagues, friends, and a wonderful and supportive community along the way. I dedicate this thesis to my parents, Dr. Pamela and Dr. Jeyaraj, and to the memory of my late grandparents, Dr. Samuel Kingsley and Mrs. Leelavathy Kingsley, whose dedication, love, and encouragement have helped me become the person I am today.

I'd like to express my sincere gratitude to Dr. Susan Liscouët-Hanke for the opportunity to contribute to the fascinating world of aircraft and for her patient guidance, unwavering support, and uplifting encouragement leading to the completion of this thesis. I would like to thank Alvaro Tamayo, Dr. Ali Tfaily, Vincent Saluzzi and Jasveer Singh at Bombardier for their expert insight, guidance and an excellent collaborative experience in developing the topics discussed in this thesis. I would also like to thank members of the AGILE 4.0 consortium for their support, especially Dr. Marco Fioriti, Dr. Jasper Bussemaker and Dr. Luca Boggero. I am immensely grateful to my wife Noémie for her steadfast support, understanding, and patience throughout the process of completing this thesis. I am thankful to Brian and Luce for their warm hospitality and encouragement, and I'm especially grateful to Brian for his meticulous proofreading of this thesis. I am grateful to my brother Danny for his constant support, encouragement, and for always being there for me.

My colleagues and friends at the Aircraft Systems Lab are truly one of a kind. I benefited immensely from their support, insight and guidance. I thank Dr. Florian Sanchez for his continuous support, encouragement, and excellent introduction to both surrogate modelling and French cuisine. I am thankful to Vijesh Mohan for his support in tool development and for helping me troubleshoot Python-related problems. I am thankful to Gala Licheva for all her support and for exemplifying a meticulous, consistent work ethic. I am grateful for all the collaborative projects I had the chance to work on during my time at Concordia; in particular, I would like to highlight the SOAPHiA project and thank Parush Bamrah, Hasti Jahanara, Mohammad Mir and Santiago Ibanez for an excellent collaborative experience. I extend my thanks to Nils Kuelper for an enriching research collaboration and to the interns Pierre-Olivier, Henry, Fabrizio and Yumna Zaheer for their support in tool development

I am grateful to Dr. Andrea Cartile for her unwavering support and encouragement, and for always offering a sympathetic ear and reassurance during the challenging moments of my PhD journey. I would also like to thank my dear friends, the boys, Akash "Sky" Sharma, Ezhil "Lisan" Shakti, Thomas "Buoy" Chacko and Paul Earnest for their support and for always being there for me, especially during the pandemic. Lastly, I thank my parents again for their endless support and sacrifices—they have been my constant source of strength and comfort during my time at Concordia.

Territorial Acknowledgment

I would like to begin by acknowledging that the research documented in this thesis was carried out at Concordia University, which is located on unceded Indigenous lands. I recognize the Kanien'kehá:ka Nation as the custodians of the lands and waters on which I conducted this work. Tiohtià:ke/Montréal is historically known as a gathering place for many First Nations, and today, it is home to a diverse population of Indigenous and other peoples. I respect the continued connections with the past, present, and future in my ongoing relationships with Indigenous and other peoples within the Montreal community.

This Territorial Acknowledgment and resources were created by Concordia University's Indigenous Directions Leadership Group (2017). To read the entire Territorial Acknowledgment and learn more about why it was written this way, please visit www.concordia.ca/indigenous/resources/territorial-acknowledgement.html.

This thesis is dedicated to the loving memory of my grandparents,
Dr. Samuel Kingsley and Mrs. Leelavathy Kingsley.

Contents

List of Figures

List of Tables

List of Abbreviations

ACE	Actuator Control Electronics
AC	Advisory Circular
AFHA	Aircraft Functional Hazard Assessment
AGILE	Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts
ARCADIA	Architecture Analysis and Design Integrated Approach
ARP	Aerospace Recommended Practices
ASSESS	Aircraft Systems Safety Assessment
ASSET	Aircraft Systems Sizing and Estimation
ASTRID	Aircraft on-board Sizing and Trade-Off Analysis
AVACON	Advanced Aircraft Concepts
CAD	Computer-Aided Drawing
CAE	Computer-Aided Engineering
CCA	Common Cause Analysis
CFR	Code of Federal Regulations
CMDO	Conceptual Multidisciplinary Design Analysis and Optimization
CPACS	Common Parametric Aircraft Configuration Schema
Ctl	Control
C	Consumer
DMDO	Detailed Multidisciplinary Design Analysis and Optimization
Dv	Device
D	Distribution
EASA	European Union Aviation Safety Agency
EBHA	Electric-Backup Hydraulic Actuator

EBS	Electrical Backup System
ECS	Environmental Control System
EDP	Engine Driven Pump
EHA	Electro Hydrostatic Actuator
EHSA	Electrohydraulic Servo Actuator
ELAC	Elevator Aileron Computer
EMA	Electromechanical Actuator
EMP	Electric Motor Pump
EPGDS	Electrical Power Generation Distribution System
ES	Energy Storage
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulations
FbW	Fly-by-Wire
FCC	Flight Control Computer
FCS	Flight Control System
FHA	Functional Hazard Assessment
FMEA	Failure Mode Effects Analysis
FMES	Failure Mode and Effects Summary
FTA	Fault Tree Assessment
HMA	Hydromechanical Actuator
HPGDS	Hydraulic Power Generation Distribution System
IDG	Integrated Drive Generator
INCOSE	International Council on Systems Engineering
L0	Level-0
L1	Level-1
L2	Level-2
LDG	Landing Gear
M1	Module-1
M2	Module-2

MB-FHA	Model-Based Functional Hazard Assessment
MBSA	Model-based Safety Assessment
MBSE	Model-based Systems Engineering
MCR	Minimum Control Requirements
Md-FHA	Model-driven Functional Hazard Assessment
MDAO	Multidisciplinary Design Analysis and Optimization
MICADO	Multidisciplinary Integrated Conceptual Aircraft Design Framework
MPGDS	Mechanical Power Generation Distribution System
MTOW	Maximum Take-off Weight
OCE	Operational Collaborative Environment
OEM	Original Equipment Manufacturer
PDS	Power Distribution System
PFCS	Primary Flight Control System Architecture
PMDO	Preliminary Multidisciplinary Design Analysis and Optimization
PMG	Permanent Magnet Generator
PRA	Particular Risk Assessment
PRIM	Primary (FCC)
PSSA	Preliminary System Safety Assessment
PTU	Power Transfer Unit
PVMT	Property Value Management Toolkit
RAAML	Risk Analysis and Assessment Modeling Language
RAMS	Reliability, Accessibility, Maintainability Safety
RAT	Ram Air Turbine
RAU	Remote Actuation Unit
RBD	Reliability Block Diagram
RCE	Remote Component Environment
REC	Reusable Elements Catalog
REU	Remote Electronic Unit
RFLP	Requirements, Functional, Logical Physical

RO	Research Objective
RPL	Replay
SAE	Society of Automotive Engineers
SEC	Spoiler Elevator Computer
SFHA	System Functional Hazard Assessment
STAMP	System-Theoretic Accident Model and Processes
STPA	System-Theoretic Process Analysis
S	Source
UERF	Uncontained Engine Rotor Failure
ZSA	Zonal Safety Assessment

Chapter 1

Introduction

The aviation industry seeks to reduce its environmental footprint and achieve net-zero carbon emissions in its operations by 2050 [1]. Canadian aviation emission reductions stemming from continued fleet renewal and the adoption of more efficient and novel aircraft are expected to contribute up to 18.8% of its 2050 targets [2]. Aircraft manufacturers are exploring electrified and novel aircraft concepts that incorporate advanced propulsion technologies such as hybrid and distributed electric propulsion. These concepts have the potential to reduce emissions significantly and meet the needs of the industry [3–8]. In addition to new aircraft configurations, the aerospace community is developing novel methods and tools to advance these new concepts [9–15].

Safety is paramount in aviation, and new aircraft concepts need to be at least as safe as present-day aircraft. As a result, is integral to the feasibility of new aircraft concepts. Novel systems technologies and increased integration between formerly isolated aircraft functions are drivers of complexity and are key areas where safety needs to be demonstrated. Aircraft original equipment manufacturers (OEMs) need to assess safety early in the design process to prevent expensive redesign, cost overruns and the risk of a promising concept being deemed unfeasible as a result of detailed safety analysis during the formal safety assessment process [16, p. 2], [17, p. 8]. Thus, the current development process needs to be adapted to integrate safety assessment at the earliest stage of design, i.e., conceptual design.

This thesis is positioned at the intersection of aircraft design, safety assessment, and aircraft systems architecting. It examines current industry practices and outlines a means of performing safety assessment early by adapting formal safety assessment methods to the level of detail typically present in conceptual design. This introductory chapter positions aircraft systems architecting within the aircraft design process and then develops the motivation for research into how safety assessment can be incorporated into conceptual design.

1.1 Background

The aircraft design process consists of three phases: conceptual, preliminary, and detailed design [18]. In the conceptual design phase, initial feasibility studies are performed that estimate important aircraft characteristics such as Maximum Take-off Weight (MTOW), fuel burn, and operating cost. Although the level of granularity in the design of the aircraft may be low, the breadth of design freedom is substantial. Aircraft designers often use multidisciplinary analysis to determine the key performance parameters of the aircraft, perform trade-off studies and finally freeze the configuration of the aircraft for entry into preliminary design.

Typically, in conceptual design, the aircraft systems are not considered in detail. Their impact is often bundled into empirical or system-specific weight estimates as shown in [18, p. 574]. Aircraft designers often manage systems architecting complexity by adapting existing architectures from well-studied baseline aircraft or past aircraft programs. Although physics-based methods for estimating the weight and power demand of systems are now publicly available and will be discussed in section 2.1.3, a broad design space exploration of different systems architectures is still difficult, as promising architectures from an aircraft-level perspective may not be feasible from a safety and reliability perspective. This is important as system safety and reliability are crucial factors in the certification potential and overall feasibility of an aircraft concept and need to be considered within the multidisciplinary studies performed during conceptual design.

Liscouët-Hanke defines aircraft systems as "those parts of the aircraft dedicated to ensuring its performance, safety, controllability, and comfort" while highlighting the aircraft power systems as being characterized by power consumption over the threshold of 1 kW [19]. Aircraft systems such as flight control, environmental control, and hydraulic and electrical systems play an important role in the safe operation of an aircraft. The architecting of these systems requires an understanding of complex interrelationships between different systems and between components within a system. Maier and Rechtin [20] define systems architecting as “...the art and science of creating and building complex systems. That part of systems development most concerned with scoping, structuring, and certification”. Fundamentally, systems architecting is a decision-making process that leads to the definition of a system architecture and its sub-components. These decisions are often based on quantitative and qualitative heuristics, some of which may be derived from the designer’s or systems architect’s experience or from best practices. Ultimately, these architecting decisions determine the feasibility, readiness, and ability of the resulting system architecture to meet the overall requirements; they have an important impact on the feasibility of an aircraft concept.

The safe and reliable operation of aircraft is the outcome of a rigorous aircraft safety assessment process carried out by large interdisciplinary teams during the aircraft design process. One such process is outlined in the SAE ARP4761 [21] (now revised as ARP4761A [22]) standard and involves significant development efforts to identify, classify, and mitigate failure scenarios and quantitatively establish that an aircraft and its systems are safe to operate. The safety of future aircraft, which are characterized by novel and complex system architectures such as hybrid-electric, distributed electric, and hydrogen-powered propulsion, as well as electrified systems and energy sources, needs to be equivalent to or better than that of their conventional counterparts. Understanding the impact of safety and certification regulations on the design of these aircraft early in the development process is crucial to establishing an efficient development process for novel aircraft.

Since the traditional safety assessment process is complex and time-consuming, it is intractable to analyze a broad design space of candidate system architectures at the same time. Incremental developments in the introduction of new technologies and systems are typically the preferred strategy as they reduce developmental risk. This is further supported by the fact that the demonstration of similarity is an acceptable means of compliance with critical certification regulations [23]. This developmental strategy works well for aircraft and systems that are well-understood or regarded as conventional. Although the present approach has also been successfully applied to the introduction of new technologies such as more-electric aircraft systems[1], the general benefits of performing safety assessment activities in conceptual design, such as improved confidence in the aircraft and systems design, reduction of risk of rework and the ability to make safety-informed architecting decisions, are indeed desirable for both current and future aircraft. More importantly, the non-trivial problem of exploring a broad design space of systems architectures early in the design and determining which ones are promising from a safety perspective holds relevance for both conventional and novel aircraft concepts. The size of the design space can vary depending on the system architecture. For an aircraft flight control and actuation system, Bauer et al. [25, p. 1026] show that the number of candidate architectures can be as large as 10^{22}. Therefore, a robust and efficient approach which applies safety assessment methods to evaluate a large design space of conventional or novel candidate system architectures is required.

The aircraft development process is part of a large organizational undertaking that integrates many disciplinary analyses and information from diverse sources. A formal systems engineering process is used to manage the developmental and design complexity involved in developing an aircraft. The systems engineering process and safety assessment process are also interrelated. Systems engineering activities define and formalize requirements that will need to be addressed by safety assessment activities. The rationale behind design decisions, analyses, and the results thereof are interconnected and documented during this process. Therefore, enabling traceability in design decisions, systems design, and analysis throughout different stages of aircraft development is important, and efficiency in this area is desirable [26]. This becomes more important as aircraft OEMs further develop their digital engineering processes [27]. Thus, any early integration of safety assessment in conceptual design will also require compatibility with downstream systems engineering and safety assessment processes, particularly in the transfer of information between different safety analyses.

The challenge in introducing safety assessment during systems architecting in conceptual design lies in balancing the required level of system architecture detail required by multiple processes. This involves meeting the system architect's perspective, i.e. the exploration of a large design space of system architectures for safety, while ensuring that the level of granularity in system architecture description in conceptual design can still be amenable for meaningful insights into safety. In addition, the incorporation of specific analyses described by the ARP4761A needs to be adapted to meet the level of granularity present in conceptual design. Finally, the shift to a digital engineering process requires that design decisions and the results of safety assessment be captured and made available for downstream design and further safety assessment activities.

1.2 Motivation for Research

This research is motivated by the needs of the industry from the perspective of an aircraft OEM as well as from an academic perspective, especially in the development of new methods to capture and evaluate system architecture information. A brief description of each of these needs is as follows:

Industry

[1] These are systems where electrical power is used to replace traditional aircraft hydraulic and pneumatic power systems [24].

Typically, the systems architecting process in conceptual design involves several steps and requires a manual exchange of information between each step, as shown in figure 1.1. A subset of system architectures derived from a known or existing architecture are evaluated for their impact on the performance of the aircraft.

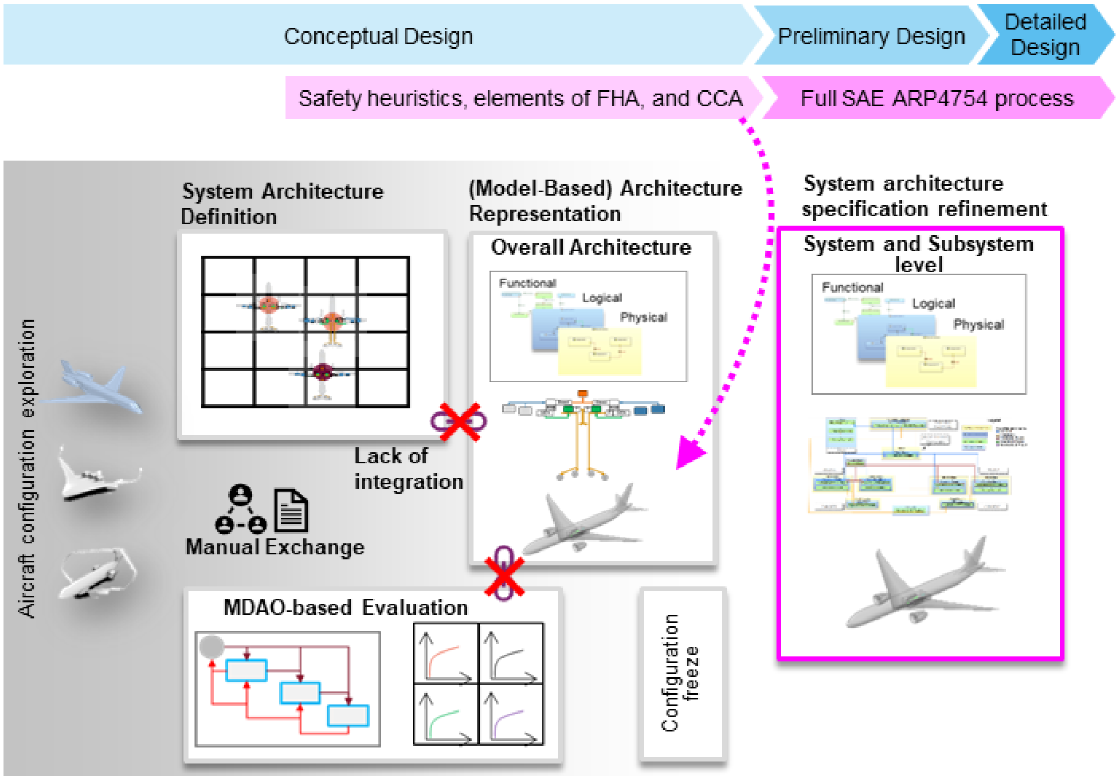


Figure 1.1: Typical system architecture design space exploration process

Aircraft OEMs use multidisciplinary analysis and optimization (MDAO) methods to design aircraft and determine if aircraft configurations meet overall requirements. Typical disciplinary analyses inside an MDAO workflow include Aerodynamics, Propulsion, Weights, and Structures. OEMs have started to integrate new disciplines, such as thermal analysis and maintainability analysis, as these become important considerations for highly integrated novel aircraft.

Conceptual systems architecting in the industry is performed in close collaboration with various design departments, including advanced design, systems, and safety. As such, an experienced system architect already incorporates some implicit design rules dealing with technological compatibility in system components, redundancy requirements for aircraft power systems, and practical system sizing considerations while defining the system architecture. However, a system architect and a safety expert will have to interact and have several architecture iterations before a system architecture is ready for evaluation. This is especially the case for novel or unconventional system architectures, such as hybrid or distributed electric propulsion system architectures, where a function such as yaw control can be shared between different systems, for example, achieved using a conventional rudder and by controlling the thrust of the distributed propulsion units. This involves both the flight control actuation system and the propulsion system, which further impacts the electrical system architecture.

Ensuring that these novel system architectures meet safety requirements, especially for critical functions, requires the architecting of the system to meet basic redundancy requirements, especially those driven by the key failure conditions, which can be different from the ones typically encountered in conventional aircraft. Additionally, safety requirements can impact the feasibility of aircraft concepts due to the weight penalties associated with redundant systems and components. This is increasingly important for hybrid and distributed electric systems since small or commuter aircraft certified under the Federal Aviation Regulations (FAR) Part 23 category are promising candidates for hybridization and electrification. However, the additional safety requirements driven by the electrification of critical functions can result in additional weight penalties that may render the aircraft unable to meet the requirements of Part 23 and thereby require re-certification under Part 25, which could ultimately render a promising concept unfeasible.

Thus, from an industry perspective, the formalization of implicit design rules and the ability to evaluate a large design space of architectures to identify promising candidate architectures that meet safety requirements are desirable, especially in the conceptual design stage. Furthermore, providing the systems architect with the tools required to perform early safety checks will help reduce the amount of interaction required before an architecture is selected for detailed evaluation. By combining early safety checks with the evaluation of a broad design space, the system architect is able to make safety-driven decisions early in the design process and enable entry into valuable design discussions on a formalized system architecture.

Aircraft OEMs are also exploring the use of digital engineering techniques such as Model-Based Systems Engineering (MBSE) to integrate traditional paper-based systems engineering activities within a digital environment. This improves traceability through different stages of design and supports analyses based on digital models. Depending on the implementation, some of these analyses, such as Model-Based Safety Assessment (MBSA), can have models that are derived from a central reference model but could evolve independently through the development process. However, the level of detail in the specification of the central model to support meaningful safety assessment using MBSA could require significant work and has the potential to not be compatible with the level of system architecture information that is available at the conceptual design stage.

The industrial perspective lies along the following three axes:

1. Incorporating safety assessment as a discipline into an industrial MDAO environment

2. Integrating MBSE methods into conceptual aircraft design

3. Integrating MBSA methods to enable safety assessment to be performed using system models in conceptual design

These perspectives are investigated in the framework of two collaborative industry projects. The first is the MDAO NextGen [28] project, which is a collaboration between the Aircraft Systems Laboratory at Concordia University and Bombardier. The second is the European Horizon 2020 AGILE 4.0 project [9], which is a cross-organizational collaborative project aimed at developing digital methods for aircraft development. This thesis documents some of the results of the research performed as part of both projects and directly addresses the aforementioned industrial research perspectives.

Academia

From an academic perspective, the design space exploration problem is interesting to study, especially when considering that architectures in the design space that meet performance requirements may be rendered unfeasible by not meeting safety requirements. An analysis of system safety and feasibility at the conceptual stage requires the description and representation of system architectures early in conceptual design. This requires balancing the level of granularity based on the available information in conceptual design with the amount of detail required to enable meaningful safety assessment activities to be performed on the architecture, which typically requires higher granularity.

The overall integration of safety and feasibility analysis into the model-based aircraft development process also requires an effective means of information transfer. Again, there are challenges with managing the appropriate level of detail to perform early safety assessment while enabling the transfer of information to build formal model-based specifications that can support more detailed assessment activities. Finally, enabling a digital thread in the aircraft development process requires methods of managing granularity across each step of systems architecting; the transfer of information from conceptual-level architecture representation to more formal model-based systems engineering specifications also needs to be developed.

Overall, this thesis uses the academic and industry needs alongside the current state of the art to formalize the following research gaps that need to be addressed:

1. **Research Objective 1 (RO1)**: Develop a methodology to effectively analyze a large design space of architecture variants

2. **Research Objective 2 (RO2)**: Establish a means of evaluating architecture feasibility by integrating certification and safety analysis early in the design process

3. **Research Objective 3 (RO3)**: Integrate new capabilities such as early safety assessment and model-based architecture representation into an aircraft MDAO environment

1.3 Organization of the Thesis

This thesis is organized as follows: Chapter 2 provides an overview of the state of the art in the field of systems architecting, safety assessment, MBSE, and MBSA. It positions the proposed safety-focused systems architecting framework in the context of other work in the literature and introduces the research questions and hypotheses that are investigated in this thesis. Chapter 3 introduces a safety-focused systems architecting framework for aircraft conceptual design and outlines the methodologies developed for each of the constituent elements of the proposed framework. A summary of enabling concepts and theory is provided wherever necessary. Chapter 4 treats the implementation of the methodologies presented in Chapter 3 for each step of the proposed framework. This includes the presentation of case studies using architectures of increasing complexity to demonstrate elements of the framework. Chapter 5 concludes the work and considers potential future developments.

Chapter 2

Literature Review

This section presents a summary of the state of the art in aircraft systems architecting, safety assessment, MBSE, and MBSA, as well as key developments in the incorporation of safety assessment in conceptual design.

2.1 Aircraft System Architecting

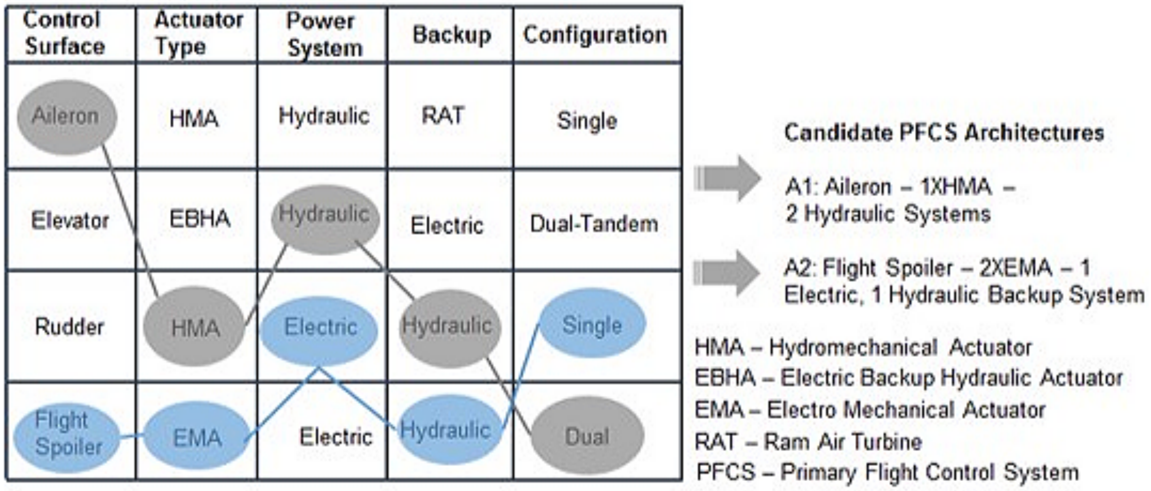
In, [29], Simmons asserts that system architecting is a decision-making process and partitions the needs of the system architect using Herbert A. Simon's [30,31] four stages of decision making for non-programmed decisions[1]. These are termed the intelligence, design, choice and review stages. The intelligence stage involves searching for the conditions that call for a decision to be made. The design stage involves formally representing the problem, structuring the representation, finding and simulating solutions, and finally viewing the results, which can be considered another form of representation. According to [32], system architecting activities in aircraft conceptual design are organized into three categories of tasks. These are system architecture definition, system architecture representation and system architecture evaluation. The system architecture definition phase aligns with Simon's intelligence phase and is envisioned to comprise choices of system architecture technology based on compilation and analysis of aircraft and system-level requirements. Some elements of system architecture definition, such as the consolidation of technology selection decisions, along with system architecture representation and evaluation, can also be considered to fall into the design activity.

In this thesis, system architecture definition is considered the synthesis of an architecture by identifying its constituent components or elements. This process includes the selection of key systems technologies, the identification of system components and the allocation of power and control to different components [33, p. 2]. The power and control architecture is particularly important from the safety perspective as this has a direct impact on the overall weight of the system and on the ability of the system to perform its function under various nominal and emergency operating conditions [19].

[1] Simmons considers non-programmed decisions as "novel, ill-structured, and often significantly consequential" citing the example of determining the mission mode of the Apollo moon landing and return mission [29, p. 22]

2.1.1 System Architecture Definition

Early approaches to system architecture design space definition used matrix-based methods to enumerate all possible system architectures. A common approach is the general morphological analysis proposed by Zwicky [34]. This approach uses a morphological matrix that classifies different architectural components and selects one option per category to create an individual architecture. Figure 2.1 shows an example of a morphological matrix for a Primary Flight Control System (PFCS) architecture. Selecting a choice under each category results in a specific architecture while parsing through all combinatorial options will populate the complete architectural design space.



Control Surface	Actuator Type	Power System	Backup	Configuration
Aileron	HMA	Hydraulic	RAT	Single
Elevator	EBHA	Hydraulic	Electric	Dual-Tandem
Rudder	HMA	Electric	Hydraulic	Single
Flight Spoiler	EMA	Electric	Hydraulic	Dual

Candidate PFCS Architectures

A1: Aileron – 1XHMA – 2 Hydraulic Systems

A2: Flight Spoiler – 2XEMA – 1 Electric, 1 Hydraulic Backup System

HMA – Hydromechanical Actuator
EBHA – Electric Backup Hydraulic Actuator
EMA – Electro Mechanical Actuator
RAT – Ram Air Turbine
PFCS – Primary Flight Control System

Figure 2.1: Morphological matrix for notional flight control system architecture showing the definition of candidate architectures from [32]

A drawback of this approach is that it does not preclude the generation of incompatible architectures. The interactive and re-configurable matrix of alternatives developed by Engler et al. [35], solve this problem by eliminating incompatible options in other categories when a certain architectural option is selected. Since the set of enumerated options grows with the number of categories under each system element, computational methods have been applied to enumerate and identify feasible architectures as shown in [36, 37].

The functionality of system architecture components introduces additional complexity to system architecture definition. Conventional system architectures present an a-priori allocation of functions to different subsystems and system architecture elements. For example, in a conventional tube-wing aircraft, the system architect is aware that the ailerons will be used to fulfill the function of lateral control. However, with novel aircraft, multi-functionality starts to emerge as a key characteristic and source of systems complexity; architectures featuring multi-functionality may lie outside the traditional system architecture design space [32, p. 22]. The aforementioned combinatorial or enumerative approaches are constrained by the choice of technology and lack of broad allocation of functionality to system components [32].

The aircraft development process follows the prescriptions of the ARP 4754 [38], which emphasizes a function-based approach to aircraft development. Additionally, the SAE ARP4761 (now ARP4761A [22]), on which the formal safety assessment process is based, is heavily function-driven. As a result, function-based architecture definition methods are desirable to manage overall design complexity.

Armstrong et al. [39, 40] present the concept of functional induction in the Adaptive Reconfigurable Matrix of Alternatives to add flexibility to architecture definition. Selecting an option to satisfy a function can lead to additional induced functions, which further require a decision on an architecture element to fulfill that function. Liscouët-Hanke introduces a function-based approach to define possible configurations for the aircraft power system architecture in [19, 41]. Here, functions are decomposed and allocated to a particular system. The architect may choose the type of power that the system will use, which then further induces the type of power that will be used in the solution that implements the power generation and power distribution functions. Lammering [42], Bornholdt [43,44], Chakraborty et al. [45] and Lampl [46–50] have also developed function-based approaches to define and evaluate system architectures. One aspect of Lampl's approach is that it incorporates multi-functionality in the definition of flight control system architectures.

Recent work by Bussemaker et al. [51, 52] has improved function-based architecture definition by introducing the Architecture Design Space Graph (ADSG). Here, the architecture is modelled as a directed graph with the nodes representing functions or elements of the architecture and the edges between the nodes capturing relationships between elements. Architectural decisions, such as the choice of component to implement a particular function, are also captured and used to build a combinatorial design space of candidate system architectures.

System architecture definition has benefited from the development of the aforementioned methods that enable the generation of robust function-based design spaces. Most methods are able to model the design space while considering architecture variants that arise when multiple solutions to a particular function are considered. This can lead to the identification of promising candidate architectures. However, most of these methods are geared towards the end goal of filtering the design space by evaluating performance and weight metrics. There are two complications that arise when following this approach. The first is that searching through a design space based on performance evaluation alone can be intractable and may not be time-efficient. Secondly, there is a risk that a promising architecture from an aircraft-level weight, fuel-burn and performance perspective could be found to be unfeasible from a safety perspective at a later design stage. Thus, the feasibility of the architecture, particularly from a safety perspective, needs to be explicitly considered and is not currently well considered in system architecture definition frameworks.

2.1.2 System Architecture Representation

System architecture representation pertains to the capture of system architecture information and presentation of the system architecture for the purpose of specifying system interfaces and communicating design decisions. A system architecture representation model identifies system components and specifies both internal and external interfaces. A representation model is a means of documenting and communicating system architecture information with different stakeholders in a system development effort [20, p. 222]. System architecture representation can encompass both written or narrative elements along with graphical elements describing system components and interfaces [53, p. 700]. Some other desirable characteristics of system architecture representation models include the representation of multiple views of the architecture, easily understandable and usable across different design phases, and the ability to capture enough information about the architecture to support the development of behaviour models [53, p. 704].

The system architecture representation model can be projected into different sub-models that focus on specific aspects of the architecture. These are called views and are further classified according to the key attributes that the view displays [20, pp. 223-225]. Architecture representation models typically exhibit data, behaviour and form views.

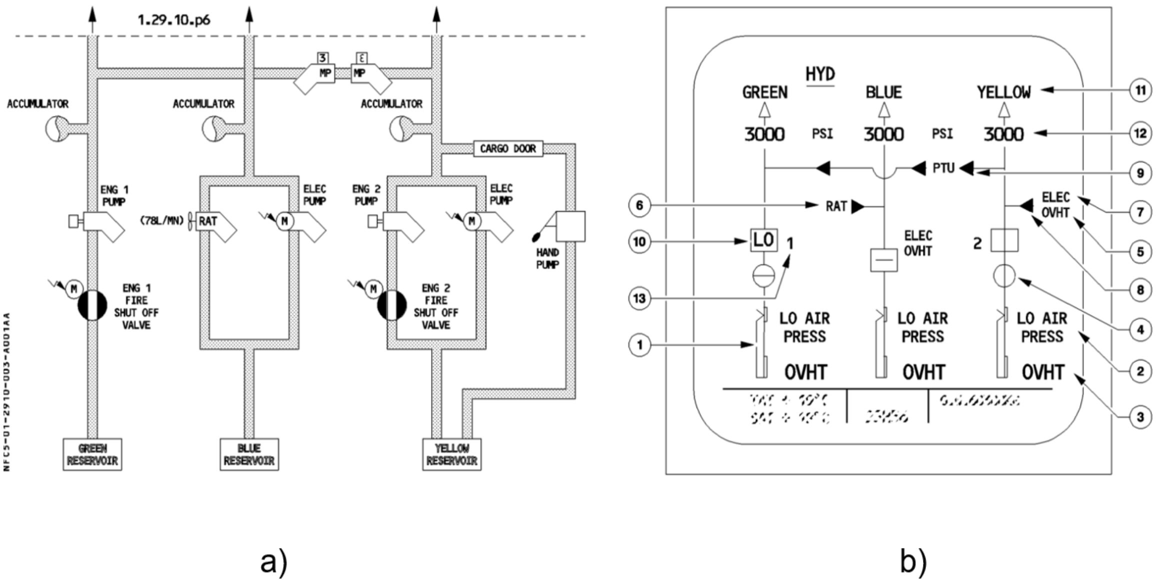


Figure 2.2: a) Representation of the hydraulic system architecture of the Airbus A320 aircraft. b) Synoptic page representation of the hydraulic system of the A320 aircraft - from [54]

Typical examples of architecture representation models or artifacts include two-dimensional drawings/schematics [54, p. 6], three-dimensional models, and interface control documentation [55]. Recent work by Fuchs et al. [56], demonstrate virtual reality based representation of cabin systems that are automatically built using information from formal specification models defined in an MBSE environment.

System architecture representation serves as an important link and means of information transfer between each stage of systems architecting. The system architect synthesizes an architecture during the architecture definition stage and provides information about the architecture to the architecture evaluation tools or formalizes the architecture for communication to other design teams. From an industry perspective, interface management artifacts such as interface control documents, [57, p. 136], and formal specification models are used to define subsystem interface requirements [58–61], allocate subsystem developmental responsibilities and inform requirements that are provided to system suppliers [62, 63].

Figure 2.2 a) shows a schematic representing the hydraulic system architecture for the Airbus A320 aircraft. This type of schematic is often found in flight crew operating and maintenance manuals. The schematic is detailed and shows the engine-driven pumps that pressurize each hydraulic system; it also shows other hydraulic components, such as the Power Transfer Unit (PTU) and Electric Motor Pump (EMP). Here, the flows of power are clearly indicated as originating from the aircraft's engines through the engine-driven pump. The allocation of hydraulic systems to different hydraulic consumers, such as flight control actuation, is represented clearly, albeit in another diagram. Figure 2.2 b) shows the representation used as a synoptic page on the A320 aircraft. Often, different diagrammatic viewpoints are required to capture power flow, power allocation and signal flow for the same system architecture.

The type of information captured in architecture representation can include the technology of system components, the number of components, the allocation of system components to elements of the aircraft, the type of signal and power flows to and from system components, and the interactions between system components. The fidelity of the architecture representation is important, as different levels of detail are required at each stage of systems architecting.

A typical example is the fly-by-wire flight control actuation system architecture, wherein detailed schematics are needed to represent power flows of electric or hydraulic power to the flight control actuators. These diagrams show the electrical power flow information for flight control computers and remote electronic units. The level of granularity in system architecture information of such schematics is high; however, representing the architecture often requires multiple diagrams. Thus, from a conceptual design point of view, it may not be possible to have all the information to create these diagrams, and for systems architecting, it may be intractable to generate these diagrams for a large design space of candidate system architectures. Therefore, the benefits of these diagrams in capturing the system architecture details for safety assessment are outweighed by the inability to use them effectively in conceptual design.

The conceptual Multidisciplinary Analysis and Optimization workflows used by aircraft OEMs to assess the overall impact of aircraft systems architecture represent system architectures using descriptors. A system architecture descriptor is a means of capturing information about the architecture that is pertinent to the sizing and analysis of the system within a tool or an MDAO workflow. However, in general, architecture description is said to encompass the synthesis of details about the system of interest, the context of the system, its elements, pertinent requirements and information about stakeholders [64, p. 60]. In this thesis, architecture description is dealt with in the context of capturing system architecture details to inform system sizing and evaluation tools. Descriptors can vary in the level of detail from simple text-based descriptors to more complex design schema [14,65]. Depending on the fidelity of the systems sizing methods, system architecture information can be codified into simple descriptions of the type of technology employed for the power system architecture or the system under study to more detailed representations of the number of system components and the interconnections between system components.

In [32, p. 56], Jeyaraj outlines the desired characteristics of system architecture representation artifacts in conceptual design as being clear and unambiguous, modular, and systematically defined. Furthermore, these artifacts are to be extensible, i.e. to be generic enough to represent new systems technologies and to capture the architecture at multiple levels of granularity. Important information that needs to be captured from the safety assessment perspective are component types and redundancies, interconnections representing the type of power, and signal flows between components. The allocation of sub-components to different parent components is also important.

Chakraborty and Mavris use a text-based descriptor that provides system architecture technology and configuration information to their systems sizing and evaluation process [45]. However, the architectures are configured according to technology buckets, with each character in the descriptor representing a particular technology. The Common Parametric Aircraft Configuration Schema (CPACS) has also been used to represent system architectures by adding custom fields supported by CPACS in the XML format [66]. These do not necessarily take into account the detailed allocation of power generation systems to power sources, although there is some flexibility in defining these aspects. Finally, some architecture descriptors in industry are implemented in spreadsheets which also contain information that is needed for system sizing methods used in architecture evaluation and sizing workflows.

A more detailed representation of system architecture can be found using system architecture specifications in a Model-Based Systems Engineering environment. Methods of architecture specification in MBSE for conceptual design are discussed by Jeyaraj [32], Krupa [67], and Tabesh [68], respectively. There are benefits to having the architecture represented as an MBSE specification model, such as improved traceability in system development by capturing the system architecture early in the design process. Although the models developed in the aforementioned approaches are tailored to conceptual design and have a low level of granularity, it can still be difficult to create these models for a large design space of candidate system architectures. The main drawback is that the specification model must be manually configured to represent it in an MBSE environment which is challenging to implement when dealing with a large system architecture design space.

Comparing the approaches to architecture description in literature against the aforementioned desired characteristics reveals that the level of detail captured in the descriptors is not sufficient. For instance, ensuring clarity in the descriptor means that the interactions between system components, especially in the flow of power and control, must be clear. The text-based description approach is difficult to parse from a visual perspective but also from the fact that it uses a codified means of description. The approach using formalized schema such as CPACS is extensible but requires additional effort to define a custom means of representing interactions between system architecture elements. The MBSE specification models are well-defined, extensible and can encapsulate information about components and associated sub-components but using them to represent candidate architectures in a large design space is intractable while maintaining a high level of detail.

When viewed from a safety assessment perspective, the detailed system architecture schematics are ideal but extensive in the number of views required to represent the desired power flows, control allocation and power allocation needed to evaluate different failure scenarios. The text-based descriptors do not contain enough information to make meaningful decisions based on safety considerations. Model-based system architecture specifications can capture significant amounts of information; they are extensible and desirable to be used during the conceptual design to capture system architecture information early, as this benefits development in later design stages. Model-based architecture specifications that use function-based descriptions of architectures are beneficial from a safety assessment perspective, as the entry point into safety assessment is the functional hazard assessment. However, again, the drawback of MBSE specifications is the amount of manual work that is required, which can be a limiting factor for system architecture design space exploration in conceptual design.

Abstraction-based representation is able to simplify and selectively outline key areas of the system architecture that are relevant to the architect [64, p. 96-97]. However, at the same time, from a safety perspective too much abstraction can obfuscate views that document important safety characteristics. Identifying the level of abstraction commensurate with providing relevant information to assess system safety while balancing the risk of becoming too descriptive requires careful synthesis of abstraction artifacts. Typically, these abstract elements that represent the system architecture should capture the form and function of system architecture elements, represent the relationships between components, and are at the appropriate level of decomposition or aggregation [69, pg .38]. Ultimately, the choice of architecture descriptor to support early safety assessment must ensure that the level of abstraction is sufficient to represent the system, its power and control flow views while also capturing component redundancies and the connection logic between different components.

2.1.3 System Architecture Evaluation

System architecture evaluation methods vary according to the desired level of fidelity and complexity. Traditional approaches in industry relied heavily on data from system suppliers, past aircraft programs and other empirical approaches. Handbook methods such as those presented in Raymer [18] and Roskam [70] develop methods that are used to estimate the weight of different subsystems. The drawback of these methods is that they may be inapplicable for novel aircraft and for new systems technologies. Furthermore, most empirical or semi-empirical methods typically use correlations on aircraft MTOW or aircraft parameters which may not be valid for newer aircraft.

Exploring a large design space of system architectures also requires filtering approaches that evaluate if a candidate system architecture is well configured and merits evaluation using physics-based sizing tools. The filtering of a large design space of candidate system architectures is approached by methods in literature in several different ways. These are configurational filters, performance-based filters, general feasibility considerations, and cost considerations. Zeidner et al. [71] and Becz et al. [72] apply abstraction within a platform-based design framework to enable design space exploration by exploring the interconnections between system architecture elements. Zeidner et al. [71] further introduce the concept of configuration filtering, which is then used in conjunction with performance-based evaluation to reduce the complete design space to a set of feasible architectures.

Garriga et al. in [73] & [74] apply the architecture enumeration and evaluation method proposed by Becz et al. and Zeidner et al. using a set of feasibility constraints for primary flight control and landing gear braking system architectures. These constraints include aspects of power allocation, control allocation, and practical cost considerations, such as ensuring the same type of braking actuator is applied to each wheel. However, important aspects pertaining to safety and certification, including the nature of power sources, such as primary or backup, and the variation of minimum power source allocation and redundancy requirements, are not considered. Control elements are not treated as part of the chain of power flow for electrical braking architectures, and the allocation of power sources to these is not considered.

Judt et al. [36], [37] have presented the use of genetic algorithms to evaluate large combinatorial spaces. Although these techniques have demonstrated utility, it is important to note that aspects of safety and certifiability, which are essential in establishing feasibility, are not explicitly considered. Certification considerations have recently been addressed through a performance evaluation-based approach in the AGILE 4.0 project [75]. However, methods to filter the design space to ensure that a remaining set of system architectures are implicitly safe are yet to be developed.

The physics-based approaches that are part of performance-based filtering methods estimate the weight and power requirements of the system based on physical sizing correlations and provide a more representative evaluation of the impact of the system architecture on aircraft-level metrics such as weight and fuel burn. These approaches typically relate the relevant physical parameters of the aircraft, such as wingspan and fuselage length, to the estimation of the powered requirements for a system. Component power densities that are publicly available or other correlations based on scaling laws are then used to estimate the weight of the system. Liscouët-Hanke [19], who introduces a function-based approach to architecture definition, also defines a power-based approach as part of their overall integrated system sizing framework. Lammering [42] and Chakraborty [76] also present integrated sizing and performance estimation framework along similar lines as Liscouët-Hanke [19]. These methods are valid for both conventional and more electric system architectures but small and regional aircraft are not well covered by these methods.

System sizing methods on their own are not sufficient to provide meaningful insights into the overall performance of the aircraft. Aircraft conceptual design and conceptual systems architecting involve multiple disciplinary analyses that must be taken into consideration to make efficient design trade-offs [18]. Aircraft OEMs have traditionally used MDAO workflows that focused on disciplines such as aerodynamics, structures and performance [77,78]. These disciplines are represented using tools at multiple levels of fidelity, and OEMs such as Bombardier have formalized a framework for the integration of multiple levels of tool fidelity in their MDAO workflows [79] featuring geometry modelling and systems integration tools.

The importance of evaluating the impact of aircraft systems architecture on top-level aircraft parameters, as well as systems integration considerations, has led to the development of several multidisciplinary analysis frameworks that integrate systems sizing as a discipline. Lammering [42] incorporates system sizing and simulation methods into the Multidisciplinary Integrated Conceptual Aircraft Design Framework (MICADO). Chiesa et al. [80] integrate systems sizing into their Aircraft on-board Sizing and Trade-Off Analysis framework (ASTRID) alongside other disciplines such as propulsion structures and reliability. Junemann et al. [81] have shown the integration of overall aircraft design (OAD) and various levels of fidelity in system sizing methods applied to the architecting of all-electric and a hybrid laminar flow control system architecture within the AVACON project. The AGILE 4.0 project has developed a collaborative MDAO platform allowing the integration of multiple design disciplines [82]. Using the MDAO framework developed in the AGILE 4.0 project, Fioriti et al. [83] have studied the electrification of aircraft systems architecture for a small transport aircraft.

Building on the trend of introducing new disciplines to MDAO workflows, the AGILE 4.0 project demonstrated the integration of an aircraft certification discipline that modified the point performance sizing constraints based on the choice of certification basis as FAR Part 23 or FAR Part 25. This, along with the integration of the safety discipline into the AGILE 4.0 collaborative workflow, is part of the overall contributions of this thesis. Other disciplines added to MDAO workflows have been thermal analysis, three-dimensional geometry, maintainability and producibility. Sanchez et al. developed a thermal risk assessment approach and demonstrated how it can be integrated alongside a three-dimensional CAD modeller in an MDAO workflow in [84]. Selim et al. [85] develop a method to assess the risk of systems component installations being difficult to maintain and provide a scoring approach that can be integrated into an MDAO environment. Buonanno et al. consider producibility within their MDAO workflow and demonstrate how promising aircraft concepts from a performance and weight perspective could have a manufacturability risk associated with them [86] by linking typical manufacturing issues with key aircraft and system sizing parameters.

Up until recently, safety assessment was largely absent from integrated systems sizing and performance estimation methods, with several efforts focusing on formally integrating safety into the systems architecting process [16, 17, 44]. These will be discussed in detail in section 2.3. Safety is also increasingly being seen as a crucial discipline to be integrated into multidisciplinary analysis and optimization. Notably, Liscouet-Hanke considers certain failure conditions for the sizing of integrated drive generators, wherein the generators are sized to take over essential loads under critical failure conditions [19]. Chakraborty et al. use safety heuristics or knowledge-driven configuration of the overall system architecture to ensure that the system architecture being provided to the estimation workflow is representative and to capture the impact of typical redundancies in power distribution and consumer systems [87]. However, both Liscouet-Hanke and Chakraborty's methods are demonstrated for conventional aircraft systems.

The need to incorporate safety becomes more important in the evaluation of novel aircraft and systems architectures, such as all-electric, hybrid-electric and distributed electric aircraft that feature increased integration and multi-functionality. The system sizing methods developed for these types of architectures do not explicitly consider safety aspects in architecture definition. Overall, the available architecture evaluation methods focus on sizing and performance estimation. These methods incorporate some safety constraints within the sizing process but do not provide any insight into the inherent safety of the architecture in terms of the response of the architecture to critical failure conditions and the severity of the impact of failures on the overall aircraft, which ultimately determines feasibility. Scalable safety assessment methods that assess a large design space of systems architectures need to be developed to allow system architects to identify promising candidates and to enable MDAO integration of safety as a discipline.

2.2 Digital Thread, Model-Based Systems Engineering and Model-Based Safety Assessment

The aircraft development process requires engineers to develop different types of artifacts that document the design of the aircraft and its systems at multiple levels of detail. These artifacts may include drawings, specifications, interface control documents or even the outcome of safety analysis such as the FHA. In a traditional development process, product life cycle management methods help manage and organize these artifacts by referencing authoritative and centralized data sources from which artifacts can be generated. Though many such systems exist and design activities are being carried out in a digital environment, such as 3D CAD and Computer Aided Engineering (CAE), there is a need to integrate all elements of the development process in a digital environment. Digital engineering seeks to do this by using authoritative sources of system information across multiple design disciplines to support development from concept to end-of-life [88, p. 2]. The use of centralized sources or information or centralized models for system development requires the information to be converted from one form to another in order to perform different design activities and analyses, forming a digital thread across different development phases.

There are several perspectives on the definition of a digital thread [89]. A digital thread defined from the perspective of systems engineering by the use of centralized system models as an authoritative source of data for system development is the one best suited to the context of this thesis. Systems engineering is moving from a traditional or paper-based approach to a Model-based systems engineering (MBSE) approach where information about the system is derived from a centralized system model. Linking MBSE to the digital thread is expected to improve the efficiency of the systems engineering process [90, p. 8]. Therefore, to improve the overall aircraft systems engineering process, there is a need to integrate the system architecting process with MBSE to benefit from the wider digital thread based development ecosystem.

2.2.1 Model-based Systems Engineering (MBSE)

The International Council on Systems Engineering (INCOSE) defines Model-Based Systems Engineering (MBSE) as "the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [91].

MBSE is a development paradigm in which a system model acts as the single source of information within the system engineering process. This central model also serves as a means of communicating the system design between disparate development teams [32]. The system model is diverse and, depending on the application can be a specification model, a schematic, a CAD model, or a simulation model [92].

MBSE has been used successfully in a diverse array of applications, from space systems engineering for interplanetary missions to the design of rail transportation systems [93–97]. More recently, Liscouët-Hanke et al. [98], Mathew et al. [99], Malone et al. [100], and Jeyaraj [32] have investigated the application of MBSE methods for the specification of aircraft systems, digital avionics networks, and flight control systems respectively. The typical utility of MBSE methods is in the clear and precise elicitation of system interactions, subsystem development responsibilities, and the development of a system architecture specification [32]. Developing such an architecture specification allows a system integrator to provide a system supplier with an exact subset of developmental responsibility that will efficiently integrate the supplier's product into the complete system.

In the AGILE 4.0 project, Ciampa et al. [101] introduced an MBSE approach to develop and evaluate MDAO workflows for complex systems. Bussemaker et al.in [51] developed a model-based approach to define and explore system architecture design spaces that are of particular interest to the system architecting process, and in [102], Bussemaker and Ciampa further explore the application of MBSE in system architecture design space exploration.

Although the potential utility of MBSE in system design and aircraft development is immense, and its industrial applications continue to rise [103], its use in system architecture representation and formal specification in aircraft conceptual design has been sparse. Model-based architecture specifications are still within the purview of later design stages when the architecture is well defined, and a specific modelling activity can be dedicated to creating a specification model [104, 105]. This is mainly due to challenges in adoption pertaining to integrating MBSE within the design process. MBSE tools and terminology present a learning curve that can contribute to resistance to adoption that also has a basis in the dynamics of the organization [106]. Furthermore, model re-usability, identifying relevant viewpoints for conceptual design, and the selection of a specific MBSE tool from the wide variety on offer is a challenge. Standardization of models at different levels of detail is an effective approach to solving these problems, but at the same time, this requires effort to develop and additional activities to establish a consensus about modelling standards within the organization or design team [32, 105].

Overall, integrating MBSE into conceptual design could enable a more effective development process as the system model is defined early and can be enriched with more information that could be pertinent at later design stages. Furthermore, simulation models and links to system sizing workflows could be supported at early stages through an MBSE specification [107]. In the context of this research, developing a model-based specification at the conceptual design stage is essential for the subsequent model-based safety assessment activity which will be explored briefly in the following section.

2.2.2 Model-based Safety Assessment (MBSA)

Model-Based Safety Assessment (MBSA) is the use of formalized models to conduct safety analyses. A model is an abstract representation of a system and contains information about system performance, behaviour, properties, and interactions between a system and its constituent components. A model used for MBSA is typically either a standalone or extended model [108]. A standalone model is a system model which also encompasses a safety model within itself. This means that the relations between different components from a safety perspective are already contained within the system model. On the other hand, an extended model is created by abstracting a system model and defining safety relations separately. Each has its advantages and disadvantages, but a standalone model can closely resemble the underlying system [108, 109].

Several approaches have been developed to conduct safety analyses using system models. Notable among these are the AltaRica modelling language developed by Point et al. [110], of which AltaRica 3.0 [111] is the latest iteration and HiP-HOPS by Papadopoulos et al. [112]. In the context of aircraft systems, Bruno et al. have developed a model-based Reliability, Accessibility, Maintainability, and Safety (RAMS) approach for conventional and more electric architectures with specific examples of model-based Failure Mode and Effects Analysis [113]. Gradel et al. have also presented a model-based safety assessment approach for conceptual design centred on a Simulink model that is then used to develop and evaluate fault trees based on the definition of failure events and assignment of safety requirements stemming from an FMES (Failure Mode and Effects Summary) conducted in-situ [114]. Gradel et al. highlight that typical MBSA methods fall outside the scope of conceptual design. They further note that for MBSA to be valuable in conceptual design, it must support the ability to rapidly define the system architecture in a graphical manner, allow re-usability of component behaviour, and support modification or fast reconfiguration of the architecture [114, p. 4].

Overall, the methods available in the literature show several formalized methods to perform MBSA and identify key criteria for meaningful application of MBSA in aircraft conceptual design. However, there still exists a need for a layered approach to integrating MBSA in conceptual design that caters to the different levels of system architecture granularity that are present in conceptual design. Furthermore, the integration of MBSA with the exploration of a large design space or the use of safety models as a tool for interactive architecting or system architecture review in conceptual design still needs to be addressed. The work presented in section 3.6.2 addresses these additional areas while meeting the criteria already established in literature for the effective integration of MBSA into aircraft conceptual design.

2.2.3 Model-based Functional Hazard Assessment (MB-FHA)

So far, the methods outlined here have been those that proposed integrated frameworks for systems architecting that also incorporate some form of safety consideration in conceptual design. In addition to these methods, there have been several approaches that tackle individual safety analysis that are prescribed by the ARP4761 standard. The FHA is one such analysis, and several methods have been developed to perform an FHA in a MBSE environment.

Conducting an FHA using system models requires consistency across the different perspectives in which a system architecture can be modelled. Maitrehenry et al. [115] highlight the importance of developing functional and operational system architecture models to support the FHA process and inform the development of subsequent safety models. Several approaches focus on architectures specified using SysML as the basis of conducting an FHA. These include an approach by Villhauer et al. [116], who demonstrate an FHA within a SysML environment using an aircraft pitch controller model as an example, and Jiang et al. [117], who implemented an FHA using SysML for the landing gear braking system test case from the ARP4761. Recent advances have been made by Jimeno et al. [118, 119]in integrating FHA and top-level requirements into the system architecting process, thereby allowing FTAs to be generated for different system architectures.

The approaches to conducting an FHA in an MBSE environment predominantly rely on building a system architecture and then analyzing it to garner insights that help populate the fields that are typically found in an FHA table. Notable among these are the approaches proposed by Lübbe, Schäfer and Lai et al. Both Lübbe et al. [120] and Schäfer et al. [121] employ an FHA-specific SysML profile extension to capture FHA information in a system model. Lübbe et al. also show the traceability between AFHA and SFHA while using a Risk Analysis and Assessment Modelling Language (RAAML) profile to develop a fault tree to support PASA activities. In another study, Lübbe et al. [122] use their Model-Based FHA approach to investigate functional similarity and re-usability between different implementations of a fuel cell system. They also showcase the capability of automatically generating visualizations that trace the hazard classifications of different functions and sub-functions for each system variant from the results of the FHA.

On the other hand, Lai et al. [123] introduce a methodology for integrating safety analysis into MBSE wherein a safety application is envisioned to generate safety artifacts from a system model containing system design and safety information. This approach emphasizes the need to reflect changes made to safety artifacts back to the system model. FHA document generation capabilities are also incorporated into this approach by using a SysML profile extension to capture FHA-related data. In [124], Lai presents a workflow that maps the interaction between system development activities and FHA activities in a proof of concept for a Model-based Functional Hazard Assessment using an aircraft landing gear extension and retraction system as an example.

Overall, the methods in the literature show how an FHA can be performed within an MBSE environment. However, the integration of these approaches with the wider systems architecting process within conceptual design, especially considering the level of granularity in system architecture specification at this design stage, still requires further exposition.

2.3 Safety Assessment in Aircraft Conceptual Design

The airworthiness regulations prescribed by regulatory bodies such as the FAA, EASA and Transport Canada need to be followed to ensure that an aircraft can be certified. These require the safety of an aircraft or system design to be demonstrated by performing different types of safety analyses. Part 25.1309 of Chapter I, subchapter C of the Title 14 Code of Federal Regulations [125] outlines the certification requirements for equipment systems and installations on transport category aircraft. Here, the regulations require that the airplane, its systems and associated components must be shown to have been designed in such a way that a single failure preventing continued safe flight is extremely improbable.

Compliance with these requirements is achieved by following a systematic safety assessment process driven by a fail-safe design approach that enshrines the principles of redundancy, isolation or independence of systems, design integrity, failure warning and indication and proven reliability, among others. The Advisory Circular AC 25.1309 [23] discusses the acceptable means of compliance with the aforementioned certification regulations and prescribes a rigorous safety assessment consisting of both quantitative and qualitative analyses to accomplish the same.

The SAE ARP 4754A [38] outlines the overall aircraft development process, and the SAE ARP4761 [21] standard formalizes the safety assessment processes that are required to demonstrate aircraft and system safety. The reader is encouraged to consult the SAE ARP4761A [22] for a detailed exposition of the aircraft safety assessment process.

On one hand, the safety assessment process requires different levels of detail in the system architecture for different types of safety analyses throughout the development process. The FHA at the aircraft and system - level are based on functional specification and at most a logical system architecture while the FMEA, ZSA and PRA require information about system component installations. The FTAs used to analyze the architecture during the SSA require a detailed system architecture specification.

On the other hand, during conceptual design, the key metrics of interest are system weight and power demands, as these are used to determine the impact of systems at the aircraft level. Furthermore, the safety assessment process, even for a single architecture, can be cumbersome with manual work and iterative analyses that need to be performed by the analysis. There also needs to be an interaction between the system architect and the safety analyst to ensure that changes are reflected in the architecture based on the outcome of the safety assessment process. In contrast, system architecture design space exploration requires the analysis of multiple architectures and the evaluation of their feasibility based on safety and performance.

The body of literature focused on introducing safety assessment into the different system architecting activities comprises the works of Chakraborty [76], Bornholdt [44], Bendarkar [17] and Jimeno [16]. These works are selected as they address a similar objective of improving the system architecting process by adapting into the conceptual design stage, analyses that are typically performed later in the design process. Chakraborty introduces an integrated system architecture sizing and performance estimation approach for architecture evaluation to determine the impact of system architectures at the aircraft level. Bornholdt addresses system architecture design space exploration and outlines a method for performing safety assessment in the conceptual design stage using reliability block diagrams. Bendarkar brings together the concept of a continuous classification of hazard severity and multi-state analysis with a Bayesian framework for component reliability assessment and decision-making to evaluate a novel aircraft configuration. Jimeno, on the other hand, introduces a framework for evaluating system architecture safety in conceptual design using the System-Theoretic Accident Model and Processes (STAMP) [126] approach as well as traditional probabilistic methods. This approach also develops the architecture using the Requirements, Functional, Logical and Physical (RFLP) [127] paradigm.

Table 2.1 compares the aforementioned approaches identified by the names of their authors against different categories of safety analyses that they support. The qualitative analysis category includes architecture analysis based on the visual representation of the system architecture that allows for a knowledge-based assessment of whether the

Table 2.1: Comparison of methods that address safety assessment in conceptual design - PC - Pre-Conceptual Design, C- Conceptual Design, P- Preliminary Design

	Type	Qualitative			Quantitative			Deployment	
	Analysis	Qualitative Architecture Analysis	Heuristics	Functional Hazard Assessment	Non SAE ARP 4761 Prescribed Methods	RBD	FTA	Level of Detail	Integration with MBSE
Method	Bornholdt [44]	✓				✓		C, P	
	Bendarkar [17]			✓	✓			C, P	
	Jimeno [16]				✓		✓	C	✓
	Lampl [50]		✓					C	
	Chakraborty [76]		✓					C, P	
	Proposed Approach	✓	✓	✓		✓	✓	PC, C	✓

architecture possesses adequate redundancy in components and power system allocations. This analysis is supported by the formalization of knowledge in heuristics or rules that an architect can use to determine if an architecture complies with safety requirements. The qualitative architecture analysis requires system architecture representation that enables the analysis of architecture safety characteristics, and the heuristics allow evaluation of the architecture against established safety rules. The Functional Hazard Assessment (FHA) is also considered within the qualitative safety analysis category as it sets safety targets and is an entry point to validation activities using quantitative safety methods. Non-SAE ARP4761 approaches encompass methods such as STPA/STAMP that are not explicitly recommended within the ARP but have been established and are gaining adoption within the aerospace industry.

The quantitative methods considered in this categorization include the reliability block diagram and the fault tree analysis. These are both included in the ARP4761 and provide quantitative insights into the system's reliability. The deployment category identifies the stage of design for which each method is suited and classifies if each method is integrated within a broader MBSE process or if it is performed in isolation.

Bornholdt's GENESYS framework supports qualitative evaluation of the system architecture as well as including a means of modelling the power distribution architecture and determining safety metrics such as overall failure rate in addition to weight and performance metrics with which to compare system architectures. From a deployment perspective, the GENESYS framework could be applied in both conceptual and preliminary design, and though the architectures are built using a function-based approach, it is not integrated within a formal MBSE environment.

Chakraborty and Marvis [87] use heuristics or knowledge that they derive from existing system architectures to configure the necessary allocation of actuators to flight control surfaces and the allocation of a minimum required redundancy in power distribution systems. The system architecture is represented using a text-based architecture descriptor and then used to size the system and determine its impact on aircraft-level weight and performance metrics. Lampl also develops the heuristics derived from Bauer et al. [25] and applies them to allocate actuators to flight control surfaces and flight control computers to different actuators in the specification of flight control system architectures. Both these approaches feature some elements of qualitative architecture analysis, but quantitative methods are not within their scope of application.

Jimeno's approach combines the STAMP method with a Fault Tree generation algorithm in an integrated systems architecting environment called AirCadia Architect, which is used to architect two system architectures. This approach is geared towards automated analysis of system architecture and not broad system architecture design space exploration. The focus is more on providing tools for the architect to specify the system architecture, set safety targets and then evaluate the architecture for safety as well as other metrics such as weight and performance. The architectures are evaluated using Fault Tree Analysis, for which an FTA generation algorithm is specified. This approach is developed for use in conceptual and features MBSE integration as it uses a custom implementation of the RFLP framework for system specification.

Bendarkar develops a new approach to tackle the uncertainty associated with system failures in novel aircraft configurations. In this approach a continuous FHA is used to provide a classification of function degradation of novel aircraft. Bendarkar developed a Bayesian framework to estimate component reliability and to evaluate multi-state failures for an aircraft based on NASA's X-57 demonstrator. Bendarkar uses a network representation to describe the power system architecture and analyzes it using FTA. This approach is suited for the early stages of preliminary design as it also relies on 6-DoF model of the aircraft to investigate different failure cases.

Considering the research objectives of this thesis, the approaches taken by Bornholdt and Jimeno to incorporating safety assessment in conceptual design fall along a similar vein. To distinguish the contributions of this thesis from the state of the art, it is necessary to examine the integration of safety assessment proposed by each method to the different systems architecting stages as a whole.

Table 2.2: Application of safety assessment to each systems architecting activity

Application of Safety Assessment per Systems Architecting Activity				Integration		Phase of Design
Method	Definition	Evaluation	Representation	Integration between Systems Architecting Activities	Applicable to Novel Aircraft / Architectures	Phases
Bornholdt	✓	✓	✓	Definition and Evaluation	Conventional/Semi-Novel	Conceptual to Preliminary
Bendarkar	✓	✓		Representation and Evaluation	Novel	Conceptual to Preliminary
Jimeno		✓		Representation and Evaluation	Conventional, Novel Architectures	Conceptual to Preliminary
Lampl	✓			Definition and Evaluation	Conventional, Semi-Novel	Conceptual
Chakraborty	✓			Definition and Evaluation	Conventional	Conceptual
Proposed Approach	✓	✓	✓	Definition, Representation and Evaluation	Conventional and Novel	Conceptual

Table 2.2 compares the stages of systems architecting to which each method applies safety assessment and also outlines the level of integration that is achieved between each stage. Chakraborty and Lampl both focus on applying heuristics to define the initial system architecture which is then provided as an input to a sizing and performance framework. However, a formalized process of deriving these heuristics is not defined. On the other hand, Jimeno's approach focuses on enabling the architect to define the architecture and then evaluate it for safety using quantitative assessment methods and make subsequent changes, if necessary, in an iterative process. Jimeno's framework was subject to industrial evaluation with positive feedback on its ability to enable the exploration of different alternatives in conceptual design, improving the understanding of how safety considerations have an impact on performance and the utility of integration of the framework into an interactive tool [p. 256] [16]. However, the evaluation also identified that the capability of modelling unconventional configurations would be an axis for improvement of the framework

in addition to reducing the amount of work required to initially set up the tool [p. 261] [16].

Bornholdt's approach also focuses on system architecture definition and evaluation using quantitative safety assessment methods. This approach also represents the architecture using reliability block diagrams and architecture schematics. Bornholdt's approach demonstrates architecture variant exploration coupled with safety analysis. However, the philosophy of the overall systems design framework in which GENESYS is a part sees the systems architecting and aircraft sizing and configuration design as different disciplines coupled with design iterations. Thus, there still exists a potential to apply safety analysis even earlier in aircraft design compared to the methods available in the literature to enable both the system architecture and aircraft configuration, sizing and performance to be developed and evaluated for safety concurrently.

Each of the aforementioned methods enables at least some form of safety analysis to be performed on a system architecture. However most of these methods consider the architecture at a level of detail commensurate to the boundary between the end of the conceptual design stage and the beginning of the preliminary stage of aircraft design. This is useful if the architecture is already known or if the variants do not deviate far from the baseline system architecture. At this stage, the aircraft configuration is fixed, and the systems architecture is developed further. However, a focus on incorporating safety analysis at this stage of design precludes several activities, such as broad design space exploration of system technologies, analysis of zonal and particular risks and overall compatibility of the system installation with the aircraft configuration. Considering the broader perspective of the system architecting process, there still exists a gap in the integrated application of safety assessment to all stages of systems architecting. There also is a gap in the integration of the systems architecting process during conceptual design with the wider aircraft development process.

Another important aspect to consider while incorporating safety analyses in conceptual systems architecting is the overall integration with the rest of the aircraft development process. All the methods shown in Table 2.2 have some degree of integration between the system architecture definition and evaluation activities. However, the activities performed in architecture definition and evaluation are performed in a silo to formal system architecture representation and the overall aircraft systems engineering process.

MBSE approaches provide a schema for formal system architecture representation. However, these are typically detailed and are developed in the preliminary design stage when the system architecture is relatively well-defined compared to the conceptual stage. Efforts have been made to enable early specification of the system architecture in conceptual design [32]. Jimeno's approach also supports the early definition of the system architecture using the Requirements – Functional – Logical – Physical (RFLP) [128] paradigm. However, bringing architecture specification using MBSE into conceptual design already requires a higher level of detail compared to what is available when the system architect first starts defining an architecture.

This results in a similar condition where only architectures that are well known and with few variants can be built using the MBSE specification thus precluding a broad design space exploration. Thus, in addition to supporting architecture specification in an MBSE environment, there is a need for a standardized system architecture representation even earlier in the design process. This representation also needs to support the activities that are performed in system architecture evaluation and safety analyses.

In summary, systems architecting and safety assessment both require the definition of the system architecture at different levels of detail. There are limitations on what types of safety assessment can be performed at which stage of design. The methods in the literature address different aspects of safety assessment, such as qualitative and quantitative approaches. Qualitative safety assessment revolves around architecture analysis, heuristics and the FHA. Quantitative safety assessment methods in literature typically use fault trees and reliability block diagrams to derive safety metrics. Model-based safety assessment methods are used to develop safety models of the system and to perform the aforementioned safety analyses as well and the use of extended models is prevalent in literature. Model-based systems engineering is being increasingly used to specify the system architecture and efforts have been made to enable architecture specification in conceptual design using MBSE.

However, the application of safety assessment to conceptual design has room for improvement along the following tracks of development.

1. Application of safety assessment to broad system architecture design space exploration using safety heuristics

2. Definition of a method to develop heuristics

3. Standardizing a means of representing system architectures commensurate with early design stage level of detail

4. Tailoring the application of ARP4761 prescribed safety analyses to system architecture at different stages of the architecture design space exploration process

5. Integration of all the stages of system architecting using a common means of information transfer and enabling the gradual detailing of a system architecture

6. Integration of zonal safety assessment and particular risk assessment into aircraft configuration definition

These areas of development identified from the literature are synthesized into the following research gaps:

- **Research Gaps 1 (RG1)**: The disconnect between systems architecting and safety assessment, especially in conceptual design, in terms of methodologies and tools

- **Research Gap 2 (RG2)**: The challenge of evaluating the safety of a large design space of potential system architecture candidates using the traditional safety assessment process

- **Research Gap 3 (RG3)**: The lack of integration between the traditional document-centric architecting process and new MDAO-based processes that are required to evaluate novel aircraft and system concepts such as hybrid-electric and distributed electric propulsion systems.

This thesis takes a systematic approach to unifying perspectives from prevailing literature and those from industrial practice. The research gaps addressed by this thesis are transformed into research questions, and each question is assigned a hypothesis. The hypotheses are then tested using a case study to determine whether they hold and to what extent they can be considered to be true.

A summary of the research questions and hypotheses for each research gap are provided below:

2.4 Research Gaps, Questions and Hypotheses

1. RG1: The disconnect between systems architecting and safety assessment, especially in conceptual design, in terms of methodologies and tools.

   Research Question 1 (RQ1): How can safety assessment be integrated into the system architecting process during aircraft conceptual design?

   Hypothesis 1: Identifying the needs of each stage of systems architecting in conceptual design, such as architecture granularity, design traceability, and link to MDAO-based evaluation, will enable the definition of an integrated framework for systems architecting that will incorporate safety assessment.

   This safety-focused system architecting framework will adapt elements of the traditional safety assessment process to the individual needs of each stage of the systems architecting process and ensure the consistent transfer of architecture information between each step of the process. Integrating elements of safety assessment into each phase of the systems architecting process raises further research questions, which are stated as follows:

   RQ 1.1: How can each stage of aircraft systems architecting, i.e., architecture definition, representation, and MDAO-based evaluation, be interlinked such that the architecture information generated during system architecture definition is transferred to other stages of conceptual systems architecting?

   Hypothesis 1.1: A graph-based system architecture descriptor can be used as a consistent medium for system architecture information storage and transfer between system architecture definition and MDAO evaluation as well as to Model-Based system architecture specification.

2. RG2: The challenge of evaluating the safety of a large design space of potential system architecture candidates using the traditional safety assessment process

   RQ2: How can safety assessment be integrated into design space exploration? Hypothesis 2: Since the traditional safety assessment process is probabilistic, it tends to drive system architectures to increased redundancy. As a result, safety heuristics for minimum required system redundancy and power allocation developed from existing architectures can be used to filter an ample design space of candidate architectures. This will enable architectures that satisfy basic requirements to be developed further in conceptual design with improved confidence in concept feasibility.

   RQ 2.1: How can a large design space of system architectures be evaluated for compliance with safety heuristics in conceptual design?

   Hypothesis 2.1: Reducing the complexity of system architecture description by representing the system architecture using a limited set of abstract components will allow a large design space of architectures to be individually described. These architecture descriptions, which are comprised of generic elements, when represented using network graphs, provide a structured framework on which the safety heuristics can be evaluated.

   RQ 2.2: How can safety heuristics be integrated into aircraft MDAO workflows?

   Safety heuristics can be integrated into MDAO based on the architecture of the implementation, i.e., collaborative and distributed or monolithic workflows. Thus, two hypotheses can be drawn as follows:

Hypothesis 2.2.1: Safety heuristics can be integrated into collaborative MDAO workflows in which the analysis tools are distributed across organizations by extracting and processing the architecture descriptor from a common information exchange schema.

Hypothesis 2.2.2: Safety heuristics can be integrated into a conventional monolithic MDAO structure using a system architecture descriptor that comprises simplified or generic abstractions of typical system architecture components.

3. RG3: The lack of integration between the traditional document-centric architecting process and new MDAO-based processes that are required to evaluate novel aircraft and system concepts such as hybrid-electric and distributed electric propulsion systems.

   RQ3: How can the traditional document-based architecting approach that is conducted within a wider document-based systems engineering process be improved to enable integration with modern MDAO methods?

   Hypothesis 3: Adopting an MBSE approach for systems architecting in conceptual design by specifying the appropriate granularity in system architecture specification can enable integration within the overall aircraft system engineering process. Furthermore, enriching system specification models with information pertinent to MDAO, extracting information by inference or explicit definition, and transferring information using a suitable medium of information exchanger can enable MDAO integration.

   RQ 3.1: How can MDAO be integrated within the aircraft systems engineering process during conceptual design?

   Hypothesis 3.1: A Model-Based Systems Engineering approach using an architecture specification framework will enable system architectures to be specified and enriched with MDAO inputs at the appropriate level of granularity to enable system architecture input to an MDAO workflow in conceptual design.

Chapter 3

Methodology: A Safety-Focused Systems Architecting Framework

This section outlines the safety-focused systems architecting framework for aircraft conceptual design that forms the core of this thesis. The proposed framework adopts elements of the safety assessment process and integrates them with the activities performed in conceptual design using a centralized representation of the system architecture to facilitate the transfer of information between each system architecting activity. The framework improves upon the state of the art in the following ways:

- Enhances the system architecture definition phase by introducing a heuristic or rule-based safety filtering method for conventional and novel system architectures (i.e., for more electric, hybrid-electric, and distributed electric aircraft). This method allows the automated extraction of feasible architectures from a large design space.

- Establishes links between the system architecture definition, the system architecture representation, and the system architecture evaluation. This specification of the links allows implementation in industry and academic environments using the principle of a system architecture descriptor which, in particular, is the missing link to executable MDAO workflows.

- Enhances the system architecture representation in an MBSE environment to ease capturing safety requirements in the system architecture earlier in the development, i.e., through linking aspects of the FHA.

This section is structured to first present the details of the methodology and then illustrate the application of each method using an example.

3.1 Safety-Focused Systems Architecting Framework

The proposed framework shown in figure 3.1 focuses on the conceptual design stage and integrates safety considerations in three specific activities. During the definition of the system architecture, the proposed framework first applies safety rules to filter through the design space of system architectures. The safety rules are derived from an analysis of certified system architectures. The main contributions here are the representation of the system architecture in a manner that enables the architecture to be checked for compliance with safety rules, the demonstration of an approach to developing safety rules, and the generalization of safety rules applicable to novel architectures.

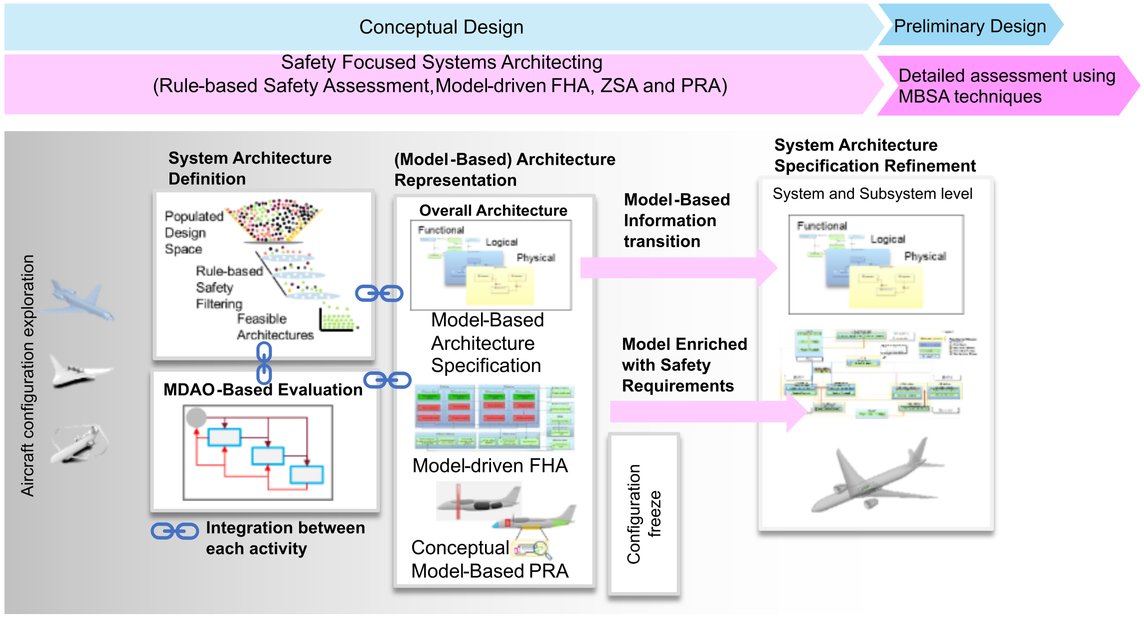
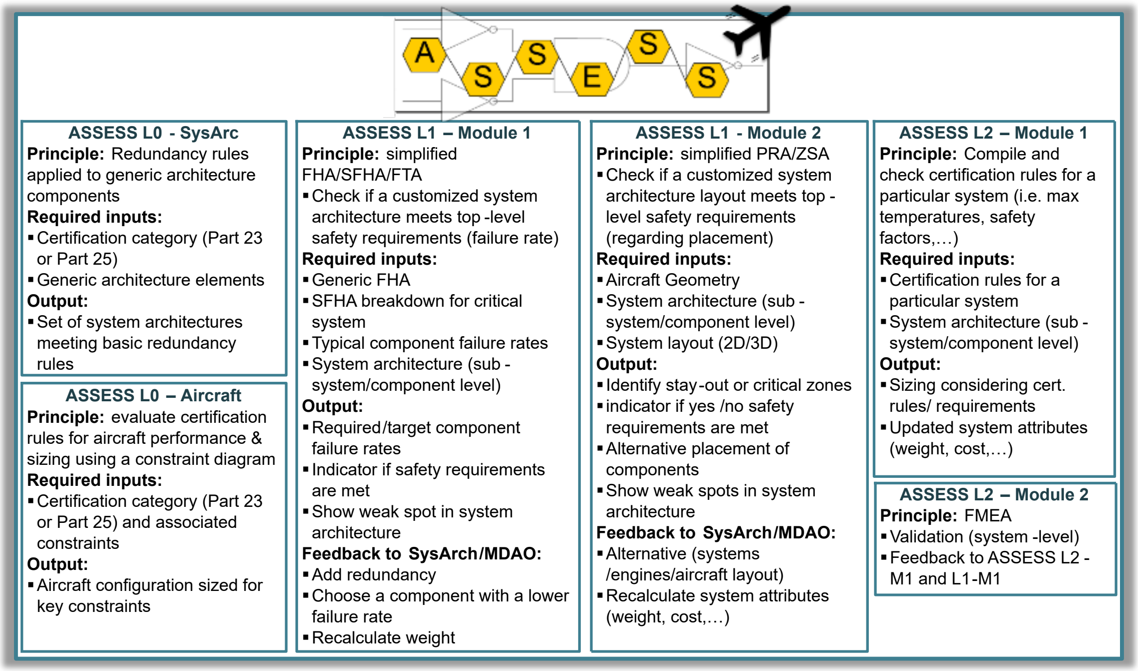

Conceptual Design
Preliminary Design
Safety Focused Systems Architecting
(Rule-based Safety Assessment,Model-driven FHA, ZSA and PRA)
Detailed assessment using MBSA techniques
Aircraft configuration exploration
System Architecture Definition
Populated Design Space
Rule-based Safety Filtering
Feasible Architectures
MDAO-Based Evaluation
Integration between each activity
(Model-Based) Architecture Representation
Overall Architecture
Functional
Logical
Physical
Model-Based Architecture Specification
Model-driven FHA
Conceptual Model-Based PRA
Model-Based Information transition
Model Enriched with Safety Requirements
Configuration freeze
System Architecture Specification Refinement
System and Subsystem level
Functional
Logical
Physical

Figure 3.1: Overview of the proposed safety-focused systems architecting framework for aircraft conceptual design

Once the design space of candidate system architectures is filtered, a reduced set of architectures that comply with the safety rules can be evaluated in an MDAO framework. The evaluation consists of sizing and performance evaluation, as well as quantitative approaches to determining system reliability. The link to the MDAO environment is established using a graph-based architecture representation and aircraft design information schema such as CPACS. Overall, this component of the framework demonstrates the integration of formal safety assessment methods into an MDAO environment.

Finally, once the remaining candidate system architectures are evaluated, the framework enables the transfer of these architectures into a formal system architecture specification in an MBSE environment. The graph-based descriptor again serves as a means of transferring information through a mapping between elements of the descriptor and components within the MBSE environment. Once the architecture is in the MBSE environment, a model-driven functional hazard assessment can be conducted, and the formal system specification model can be sequentially enriched with additional safety information and provided downstream for formal MBSA. The framework also incorporates common cause analysis within the systems architecting stage. This is performed using CAD models of system installations and carrying out zonal safety analysis and risk assessment.

The framework presented in figure 3.1 is implemented as a tool suite called Aircraft System Safety Assessment or ASSESS. ASSESS consists of a set of software modules, as shown in figure 3.2.



ASSESS L0 - SysArc
Principle: Redundancy rules applied to generic architecture components
Required inputs:
- Certification category (Part 23 or Part 25)
- Generic architecture elements

Output:
- Set of system architectures meeting basic redundancy rules

ASSESS L0 – Aircraft
Principle: evaluate certification rules for aircraft performance & sizing using a constraint diagram
Required inputs:
- Certification category (Part 23 or Part 25) and associated constraints

Output:
- Aircraft configuration sized for key constraints

ASSESS L1 – Module 1
Principle: simplified FHA/SFHA/FTA
- Check if a customized system architecture meets top -level safety requirements (failure rate)

Required inputs:
- Generic FHA
- SFHA breakdown for critical system
- Typical component failure rates
- System architecture (sub - system/component level)

Output:
- Required/target component failure rates
- Indicator if safety requirements are met
- Show weak spot in system architecture

Feedback to SysArch/MDAO:
- Add redundancy
- Choose a component with a lower failure rate
- Recalculate weight

ASSESS L1 - Module 2
Principle: simplified PRA/ZSA
- Check if a customized system architecture layout meets top - level safety requirements (regarding placement)

Required inputs:
- Aircraft Geometry
- System architecture (sub - system/component level)
- System layout (2D/3D)

Output:
- Identify stay-out or critical zones
- indicator if yes /no safety requirements are met
- Alternative placement of components
- Show weak spots in system architecture

Feedback to SysArch/MDAO:
- Alternative (systems /engines/aircraft layout)
- Recalculate system attributes (weight, cost,…)

ASSESS L2 – Module 1
Principle: Compile and check certification rules for a particular system (i.e. max temperatures, safety factors,…)
Required inputs:
- Certification rules for a particular system
- System architecture (sub - system/component level)

Output:
- Sizing considering cert. rules/ requirements
- Updated system attributes (weight, cost,…)

ASSESS L2 – Module 2
Principle: FMEA
- Validation (system -level)
- Feedback to ASSESS L2 - M1 and L1-M1

Figure 3.2: Aircraft System Safety Assessment (ASSESS) Overview

Each module shown in figure 3.2 implements a specific aspect of the safety assessment process that is integrated into systems architecting and addresses safety at an increasing level of system architecture granularity. The granularity of each module is expressed using the denotation L0, L1, and L2, with L0 being the lowest and L2 being the most detailed level of granularity, following the principles for multi-fidelity approaches in conceptual design, as established by Piperni et al. [79] and further developed for system-level analyses by Sanchez et al. [84]. Here, L0 methods of the safety assessment are suitable for conceptual design exploration purposes only. However, L1 and L2 methods are for more detailed evaluation and lead into the formal safety assessment methods of the SAE ARP4761.

The modules of ASSESS also address the above-mentioned research gaps. The ASSESS L0 module implements the rule-based safety assessment whereas ASSESS L1-M1, L1-M2, L2-M1, and L2-M2 help capture the system architecture in an MBSE environment and enable safety analyses such as FHA, PRA, Zonal Safety Assessment (ZSA), and Failure Mode and Effects Analysis (FMEA).

ASSESS-L0 analyses at the aircraft and system levels allow for early checks with a low level of granularity. This enables the traversal of large design spaces, thereby ensuring that L1 analyses can focus on a limited set of system architectures. The emerging safety properties (such as required redundancy or segregation) of the system architecture are addressed with L1 methods. These L1 methods also focus on utilizing information from the aircraft level, such as the external and internal geometry of the aircraft and the placement of system architecture components. L2 methods are applied when a single architecture (or a very limited set) remains and can be analyzed in detail to identify additional safety requirements. Downstream safety tools are then used to perform detailed safety analyses using the system model as the primary artifact. Several strategies to apply elements of the proposed framework in practice are outlined in section D.1 of appendix D.

3.2 ASSESS L0 - Aircraft - Aircraft Level Assessment

ASSESS L0 - Aircraft is an aircraft-level module that estimates the fuel burn, MTOW, component weights, and performance of an aircraft from a set of top-level aircraft parameters. Aircraft sizing is subject to performance constraints that are applied according to certification regulations obtained from the airworthiness standards for normal and transport category aircraft such as 14 CFR Part 23 [129] and Part 25 [130], respectively, depending on the category of aircraft. This module implements typical sizing methods from [131, 132], that apply to conventional as well as hybrid and all-electric aircraft. This module provides aircraft-level information such as weights, drag, and power demands to system architecture evaluation tools and also uses the weight and drag penalties from the architecture evaluation to determine the impact of the system at the aircraft level.

The certification rules pertaining to aircraft performance that have been implemented in the L0 aircraft module are shown in [75], where they have been integrated into an MDAO framework.

3.3 ASSESS L0 - SysArc - Rule-based Safety Assessment

The ASSESS L0 - SysArc implements rule-based safety filtering that enables the evaluation of a large design space of system architecture. It addresses the need to incorporate safety assessment within conceptual design by testing a large design space of candidate architectures against predefined safety rules or heuristics. Rule-based safety assessment identifies architectures that meet minimum safety requirements and improves early confidence in system architecture feasibility. The methodology to implement rule-based safety assessment in the proposed framework is based on three key components and several associated elements that enable integration with other activities in the framework, such as architecture evaluation in MDAO and the development of an MBSE specification model to support MBSA.

The three key enablers for the rule-based safety filtering of a large design space are:

1. Generic System Architecture Representation
2. Rule Identification, Formalization, Evaluation and Filtering
3. Rule Generalization and Extension to Unconventional/Novel Architectures

The generic representation of the system architecture is a key component in enabling the evaluation of a system architecture for safety at this early stage in the design process. Furthermore, the generic system architecture representation plays a crucial role in bridging different design activities, such as MDAO evaluation and MBSE specification, by facilitating the transfer of necessary information to map the architecture from one activity to another. This will be discussed in later sections.

3.3.1 Generic System Architecture Representation

At the conceptual design stage, the critical aspects of system safety are those that affect the feasibility of the system architecture. This means that any system architecture must be configured in such a way that it meets specific safety requirements while at the same time meeting performance, weight, and fuel burn requirements at the aircraft level. This relationship between system-level safety and aircraft-level performance impact is typically captured through the system weight and power requirements. From a safety perspective, system redundancy in power and control paths directly impacts the weight of the system architecture. Selecting a system architecture that may not meet redundancy requirements can lead to a scenario where the architecture will need to be reconfigured in later design stages with the risk of the additional incurred weight affecting the overall aircraft-level performance and feasibility of an aircraft concept.

System architectures are documented and studied using system architecture representation artifacts. These may be specification documents, diagrams, layouts, and different types of models. The system architect deals with these artifacts at multiple levels of granularity in the system architecture description. In order to filter a design space of architectures for safety, especially by checking for redundancy requirements and the allocation of power to different system architecture components, it is essential to represent the architecture at the appropriate level of detail to efficiently enable these studies. At the same time, the system architect also needs to be able to comprehend the architecture without excessive effort. As discussed in section 2.1.2, the system architecture representation needs to follow the key principles of clarity, conciseness, and encapsulation.

Visual representations of system architecture are particularly useful in studying the architecture. Aircraft documentation such as flight crew operating manuals, aircraft familiarization training, and pilot manuals often have schematic or pictorial representations of the system architecture. These documents discuss important failure cases and use visual representations to improve the pilot's understanding of the situation. These representations deal precisely with communicating the available redundancies in the power architecture that are allocated to a specific system.

However, visually inspecting each system architecture may be intractable for large design spaces. Furthermore, capturing aspects such as power flows and control flows may require multiple diagrams that show each perspective. Thus, a middle-ground representation is prescribed wherein useful information about the system, such as the allocation of system components, allocation of power and control, and the required redundancies in power and control, are clearly represented visually using graphical means but also in a format that can be used to transfer architecture information between different design stages.

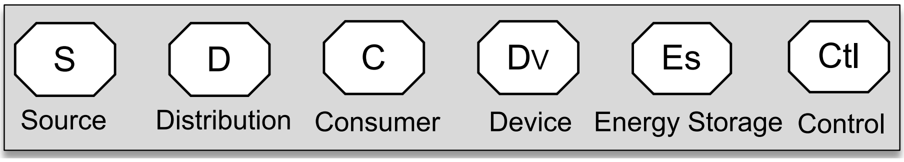


Figure 3.3: Generic elements used to represent system architectures

The generic system architecture representation presented here strikes a balance between the required level of detail in the system architecture, simplicity of visual representation, and the ability to capture architecture information. This descriptor comprises a set of generic elements named Source, Distribution, Consumer, Device, Energy Storage, and Control, as shown in figure 3.3. Each element represents an abstraction of component types and categorizations such as power generation, power distribution, and power consumption components typically found in aircraft power system architectures. These elements are classified based on their interaction with the flow of power in a system architecture. As such, they may be related to existing classifications in literature, as shown in table 3.1. The source element deals with the provision of power and is representative of power generation elements of which a prime mover is a possible physical implementation. An essential feature of these generic elements is that they may be used at different levels of granularity. For instance, the source element could be used at the system or component level, representing either a complete power generation system or a specific power generation element.

Table 3.1: Nomenclature of generic elements

Generic Element	**Referenced Nomenclature**	**Examples of Physical Components**	**Examples of Associated Elements**
Source	Power Generation Systems [19], Prime Movers [87]	Engine, Battery	Fuel
Distribution	(PDS) [19], (MPGDS, HPGDS, EPGDS) [87]	Hydraulic, Electrical Distribution System	EDP, EMP, IDG
Consumer	Power Consuming Systems [19], Subsystems (FCS, ECS, LDG) [87]	Flight Control System, Flight Control Actuators, Landing Gear Braking Devices	EHA, EMA, EHSA
Device	-	Wheels, Control Surfaces	-
Energy Storage	-	Fuel Tank, Battery, Hydrogen Storage Tank	Fuel
Control	-	Controller	REU, FCC

The "Distribution" element represents a routing of power from the source element and can be used to model power distribution systems and elements. "Consumer" elements are used to represent power-consuming systems and individual components. One can use the "Device" elements to represent passive or structural components such as wheels and control surfaces. Connections between elements are made hierarchically, starting from the source elements and proceeding to distribution, consumer, and device elements. Connections between elements of the same type are permitted, and a single element on the lower rung of the hierarchy can be connected to multiple elements at higher levels. However, a direct connection between any two elements without passing through the intermediate elements is not permitted. In addition to the aforementioned elements that are abstractions of components in aircraft power system architectures, this framework also introduces an element to represent components that affect the control of power. The control element enables the flow of power to the standard generic elements.

Consider figure 3.4, which shows a typical schematic of a hydraulic power system supplying the wheel braking system of the Airbus A320. Here, the source element represents the sources of hydraulic power, such as the engine, while the distribution element represents an independent hydraulic distribution system. The wheel-brake actuation system is represented using the consumer element, and the wheel itself is the device. The abstracted elements capture the key components that are part of power flows in the system architecture.

These power flows are important from a safety perspective as they determine if a system can still perform its function under critical failure conditions.

The generic elements presented here enable the representation of the system architecture from a secondary power flow and allocation perspective, which provides the system architect with visual insight into the redundancies in power systems, components, and power system to component allocations. With a basic representation, the architect can evaluate safety rules based on interconnections between the generic elements, study the impact of common resource failures such as the loss of a hydraulic system or an all-engine failure, and review the architecture with safety experts to determine if additional redundancies are required.

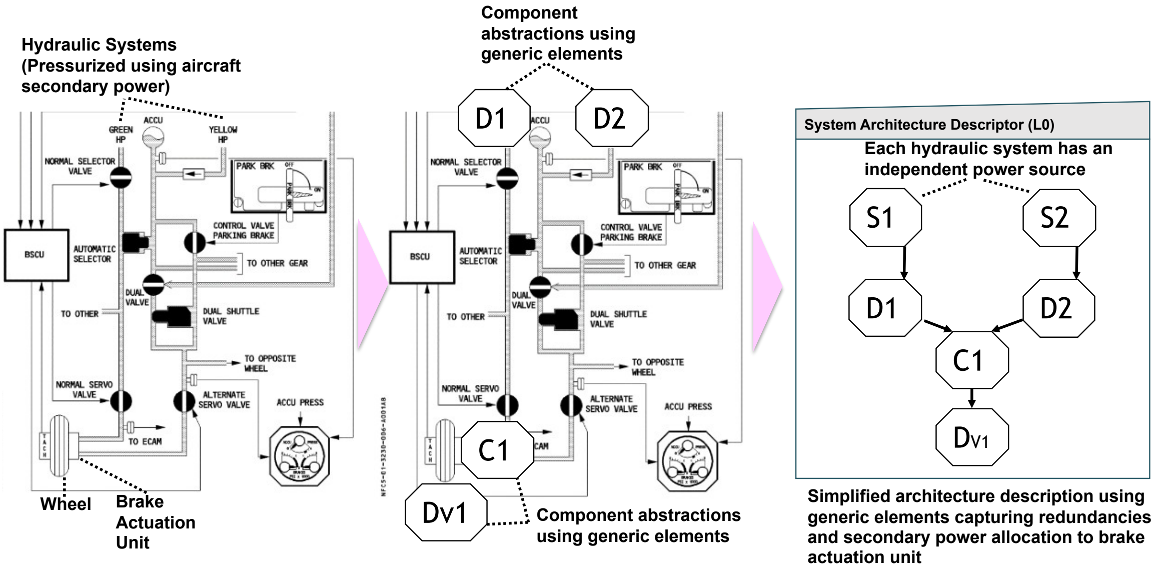

Hydraulic Systems (Pressurized using aircraft secondary power)
GREEN HP
ACCU
YELLOW HP
NORMAL SELECTOR VALVE
PARK BRK
BSCU
AUTOMATIC SELECTOR
CONTROL VALVE PARKING BRAKE
TO OTHER GEAR
DUAL VALVE
DUAL SHUTTLE VALVE
TO OTHER
TO OPPOSITE WHEEL
NORMAL SERVO VALVE
ALTERNATE SERVO VALVE
ACCU PRESS
TO ECAM
Wheel
Brake Actuation Unit
Component abstractions using generic elements
D1
D2
C1
Dv1
Component abstractions using generic elements
System Architecture Descriptor (L0)
Each hydraulic system has an independent power source
S1
S2
D1
D2
C1
Dv1
Simplified architecture description using generic elements capturing redundancies and secondary power allocation to brake actuation unit

Figure 3.4: System architecture schematic to architecture descriptor based on generic elements using the landing gear braking system for one wheel of the Airbus A320 aircraft. Underlying schematic adapted from [54].

The generic elements can be used at various levels of abstraction. Higher levels of abstraction are used when an analysis of essential redundancies and critical failure conditions needs to be performed. However, each generic element can also be assigned sub-components and is able to store component properties. For example, in figure 3.5, the source component is first illustrated at the highest level of abstraction. The information stored in the descriptor identifies the component as a hydraulic source. This level of representation captures high-level failures caused by the loss of an engine or a main source of secondary power. However, the second representation shows that the source element is allocated an engine-driven pump (EDP), which means that the loss of hydraulic power from the source element could be traced to the loss of an engine-driven pump or the failure of the source element itself. In figure 3.5, the source element is allocated both an EDP and an Integrated Drive Generator (IDG); this shows that the loss of hydraulic power could be attributed to the loss of the EDP, and the potential loss of electric power can be attributed to the loss of IDG. The loss of the entire source element will deprive any connected distribution nodes of electric and hydraulic power. Furthermore, the descriptor can be traversed to isolate the impact of the failure of sub-components such as the EDP without affecting the description of how the source element also supplies electric power to electric consumers (such as C3) using the IDG.

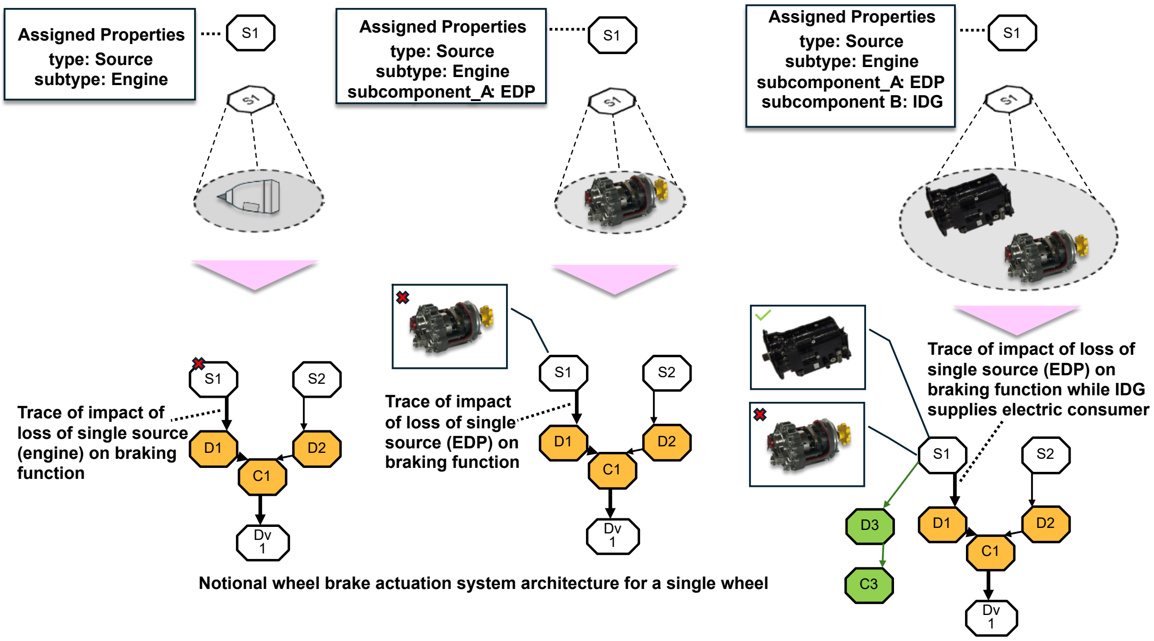

Assigned Properties
type: Source
subtype: Engine
S1
Trace of impact of loss of single source (engine) on braking function
S2
D1
D2
C1
Dv 1
Assigned Properties
type: Source
subtype: Engine
subcomponent_A: EDP
Trace of impact of loss of single source (EDP) on braking function
Assigned Properties
type: Source
subtype: Engine
subcomponent_A: EDP
subcomponent B: IDG
Trace of impact of loss of single source (EDP) on braking function while IDG supplies electric consumer
D3
C3
Notional wheel brake actuation system architecture for a single wheel

Figure 3.5: Enabling allocation of sub-components to generic elements using encapsulation within graph nodes

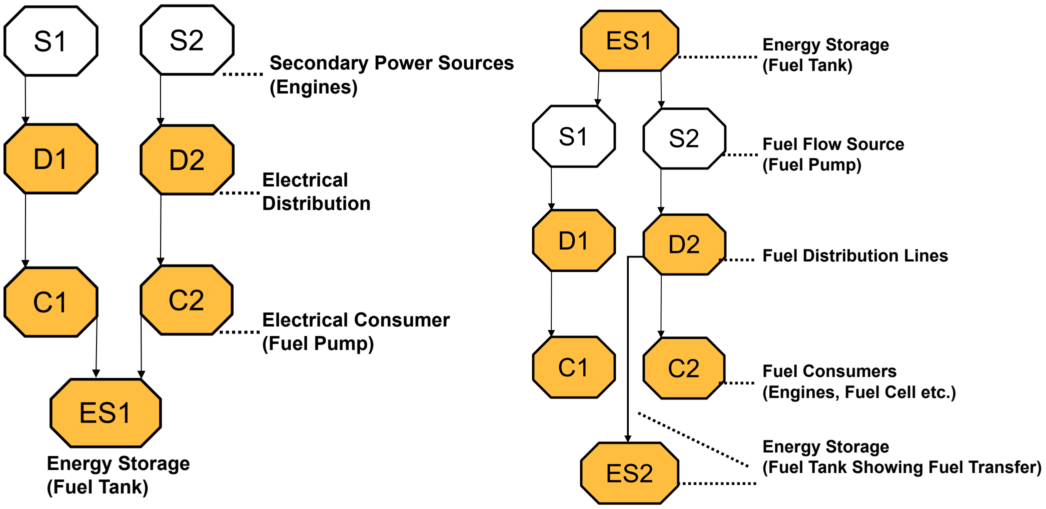
Generic elements can also encapsulate descriptions of subsystems. For example, an electrical distribution element can store a description of the components and interconnections between components in an electrical bus. Generic elements can also store properties and data associated with the allocated sub-components; for example, in the case of an EDP associated with a source element, information such as rpm and flow rate can be associated with the generic element. If a source element is set to represent an engine, then relevant parameters such as fuel flow rate, engine spool rpm, shaft off-take power, and the failure rates of the engine or specific engine components can also be stored.

The generic element descriptor is implemented as a graph object using the Networkx library in python [133]. A graph is described as a pairwise relationship between two elements [134]. A collection of element pairs can be used to describe the interrelationships between components in a system. When used for system architecture description, graphs offer a dynamic representation of the architecture and a multi-level repository for storing architectural information. At the abstract level, a graph can be used to describe relationships between components and capture flows of power, energy, and control within a system architecture. The graph-based descriptor also stores information pertinent to system architecture elements within its nodes and interconnections. Graph elements and interconnections can be queried and evaluated, which makes it suitable for testing safety rules.

Thus far, the generic elements have focused on representing power flow through the aircraft's secondary power network. However, depending on the system to which secondary power is being supplied, the flow of power and the operation of the power-consuming component is metered by other components performing a control function. For example, in an aircraft flight control actuation system, hydraulic or electrical power may be supplied to the actuators, but the operation of the actuator is determined by an input signal from the cockpit and is managed by a controller or remote electronic unit. This controller itself is supplied secondary power from the aircraft's electrical system, and thus, from a safety perspective, the flow of power and control are interrelated. These features are captured in the generic element descriptor by introducing an auxiliary element, i.e., the control element denoted by "Ctl". The control element takes as an input electrical power and provides as an output a control signal to a consumer element. This is illustrated in figure A.1 found in section A.2, which shows two types of exchanges; the first is the flow of hydraulic power from the source element through a distribution element to a consumer element. The second shows how a consumer element receives both electrical power from a distribution element and a control signal from a control element.

Additional flows, such as fuel flow, also need to be considered when representing aircraft fuel system architectures. Key elements in a typical aircraft fuel system architecture include fuel tanks, fuel pumps, piping, and fuel valves. Fuel pumps are linked to the aircraft's secondary power system through the aircraft's electrical system, so the flow of power to the fuel system needs to be captured. At the same time, the fuel system provides fuel to the engines or elements that are associated with the aircraft's primary power; from a safety perspective, it is important to understand how the aircraft fuel tanks and distribution lines are linked to each other and how they are linked to the primary power sources. This requires multiple views of the system described using the same nomenclature of elements, with the introduction of the energy storage element used to represent components such as batteries and fuel tanks.

The energy storage element, represented by "ES", provides a repository for energy in the form of fuel or chemical energy to be supplied to source elements or to be fed to the consumer elements. An example of the proposed multi-view approach is shown in figure 3.6. In figure 3.6(a), a generic element description of an aircraft fuel system shows the power view where the sources and distributions represent the secondary power generation and distribution elements, while the consumer represents fuel pumps which feed the energy storage element representing a fuel tank. Figure 3.6(b), shows the mass flow view where the fuel from the energy storage element is fed into the source element (representing a source of fuel flow such as a fuel pump), which is then fed into a fuel distribution system represented by the distribution element and finally sent to a consumer element which represents primary power sources such as engines. The terminal elements in a mass flow view for the fuel system can be either consumer elements representing engines, or a combination of consumer elements and energy storage elements, which represent fuel supply to an engine and the fuel system performing a transfer between fuel tanks, respectively.


S1
S2
Secondary Power Sources (Engines)
D1
D2
Electrical Distribution
C1
C2
Electrical Consumer (Fuel Pump)
ES1
Energy Storage (Fuel Tank)
ES1
Energy Storage (Fuel Tank)
S1
S2
Fuel Flow Source (Fuel Pump)
D1
D2
Fuel Distribution Lines
C1
C2
Fuel Consumers (Engines, Fuel Cell etc.)
Energy Storage (Fuel Tank Showing Fuel Transfer)
ES2

(a) Power flow view

(b) Mass flow view

Figure 3.6: Power flow and mass flow captured using the generic element descriptor

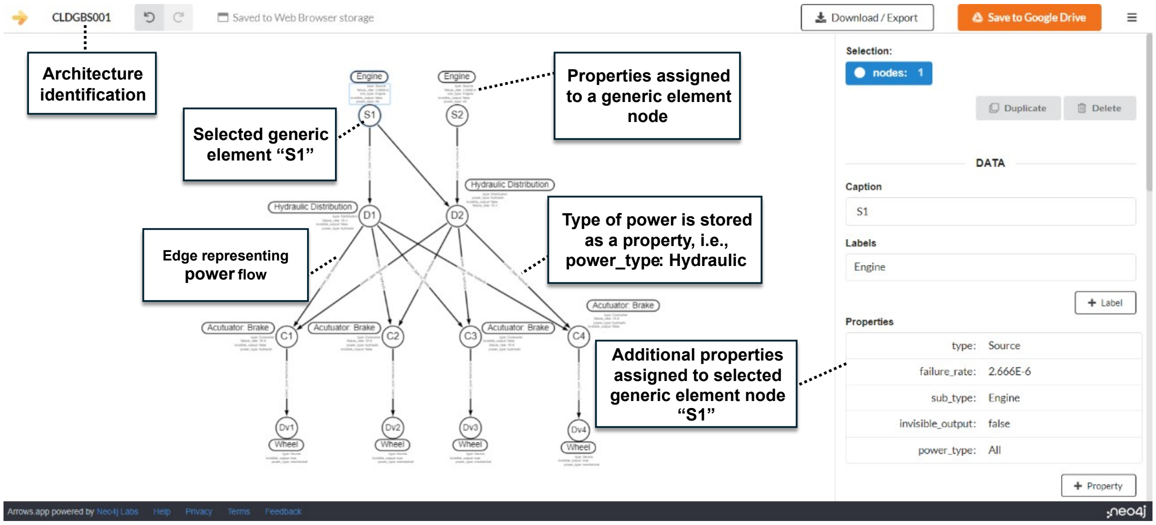
Connections between generic elements follow a prescribed scheme. Source elements can connect to distribution elements. Distribution elements can connect to each other and also to consumer elements. Consumer elements can connect to device elements, but not to each other. However, if certain architectures require the exchange of power between consumer elements, then connections can be accommodated, provided that the safety rules and the rule checking are also appropriately modified. Control elements can be connected to any element but must receive power only from a distribution element.

The generic element descriptor also enables the concurrent representation of multiple systems within a single graph. This unified representation allows the system architect to allocate power distribution to different systems and trace the impact of the loss of common resources, such as power sources or distribution elements, on multiple systems at the same time. A large-scale multi-systems specification and analysis can be performed using the generic element descriptor in the following ways:

1. Using Visual specification and analysis
2. Using graph algorithms
3. Storing and querying the architecture in a graph database

Visual specification and analysis

Depending on the implementation, the generic element descriptor can be built using programmatic means or visual means. Programmatic methods encompass generating the graph using scripts or reading an adjacency matrix or any other static representation of the graph from a file. However, having a visual method enables an interactive systems architecting process. In this work, an open-source application called Neo4j Arrows [135] is used to build the generic element descriptor. Arrows provides an interactive user interface where graph nodes can be defined and formatted. It also enables the user to connect nodes together using edges. Furthermore, each node can be assigned custom properties and can be formatted as desired. Figure 3.7 shows a notional aircraft landing gear braking system architecture being developed within the Arrows application. Here, the graph can be inspected visually to check how each power distribution system is assigned to the brake actuation units.



Architecture identification

Selected generic element "S1"

Properties assigned to a generic element node

Edge representing power flow

Type of power is stored as a property, i.e., power_type: Hydraulic

Additional properties assigned to selected generic element node "S1"

Selection: nodes: 1

DATA

Caption: S1

Labels: Engine

Properties

type:	Source
failure_rate:	2.666E-6
sub_type:	Engine
invisible_output:	false
power_type:	All

Figure 3.7: Interactive systems architecting using generic elements within the Neo4j Arrows application

Finally, to facilitate further analysis using safety rule checkers, quantitative safety assessment, and other evaluation methods, the graph can be exported from the Arrows application in several formats. Among these are the JavaScript Object Notation (JSON) and the Cypher query language. These are of particular interest as they support some of the subsequent graph analysis methods.

The graph evaluation scripts developed in this thesis are written in Python and rely on the Networkx package [133] to read, represent and apply various graph algorithms. However, since the Networkx library does not directly support importing graphs stored in the JSON format, a converter was developed to convert the JSON file into a Graph Markup Language (GML) file, which can be read by Networkx.

3.3.2 Safety Rule Identification and Formalization

Rule-based safety assessment draws on many different design approaches such as knowledge-based engineering, heuristics-based design, and the prescriptions of systems engineering and safety-driven development guidance material which includes the SAE ARP4754 and SAE ARP4761 standards. The proposed approach is based on the assumption that the traditional aircraft safety assessment process, according to the SAE ARP4761, is probabilistic and tends the architecture towards greater redundancy [136]. Furthermore, guidance material such as AC 25.1309 [23] advises a fail-safe design philosophy with the driving principles being a focus on ensuring system independence, segregation, and dissimilarity. The inverse relationship between the severity of failures and allowable probability of occurrence that is used in safety analyses that are an integral part of the safety assessment process, such as the functional hazard assessment, also contributes towards promoting greater redundancy. These aspects are key drivers towards increased redundancy in systems, subsystems, and components to meet safety requirements.

Certified aircraft system architectures have been subject to extensive safety assessment and have been designed to meet the key requirements of regulatory texts. As such, the underlying assumption of the proposed approach is that these system architectures codify the design principles enshrined in regulatory texts and the outcomes of the formal safety assessment process. Therefore, the hypothesis is that safety rules governing the allocation of power to system components and the minimum required redundancies can be extracted from an extensive study of such systems. Furthermore, this thesis posits that since novel aircraft must also be certified to the same standards, the safety rules extracted from a study of conventional system architecture can be made generic and applicable to filtering a design space of novel architectures as well.

Safety rules are derived per system by analyzing information about a particular system, such as the landing gear braking system, from various sources using both a top-down and bottom-up approach. Technical documentation such as schematics, operating manuals and descriptions of the architecture in literature are rich sources of system architecture information. Regulatory documents such as Title 14 CFR Part 25 [130] and 23 [129] contain specific guidance related to aircraft systems that affect overall system safety. Early analyses in the safety assessment process, such as functional-hazard assessment, provide information on typical failure conditions that need to be considered in systems design. Knowledge-based engineering manifests itself in the form of reviews and feedback on the formulation of rules based on industrial best practices, and the experience of safety engineers is also incorporated in the drafting of safety rules. Finally, the generic element descriptor itself is used to capture an abstraction of the systems architecture, making it more amenable to studying and identifying patterns in power and redundancy allocation.

Figure 3.8 illustrates the combination of top-down and bottom-up methods that result in the synthesis of safety rules. From a top-down perspective, the FHA can help identify key failure conditions, i.e., those that result in hazardous or catastrophic conditions. The top-down approach also considers direct guidance from regulatory texts containing an explicit definition of system architecture safety requirements. However, in some cases the regulatory text may be ambiguous or open to interpretation and further analysis of actual system architectures is required to synthesize a formal safety rule.

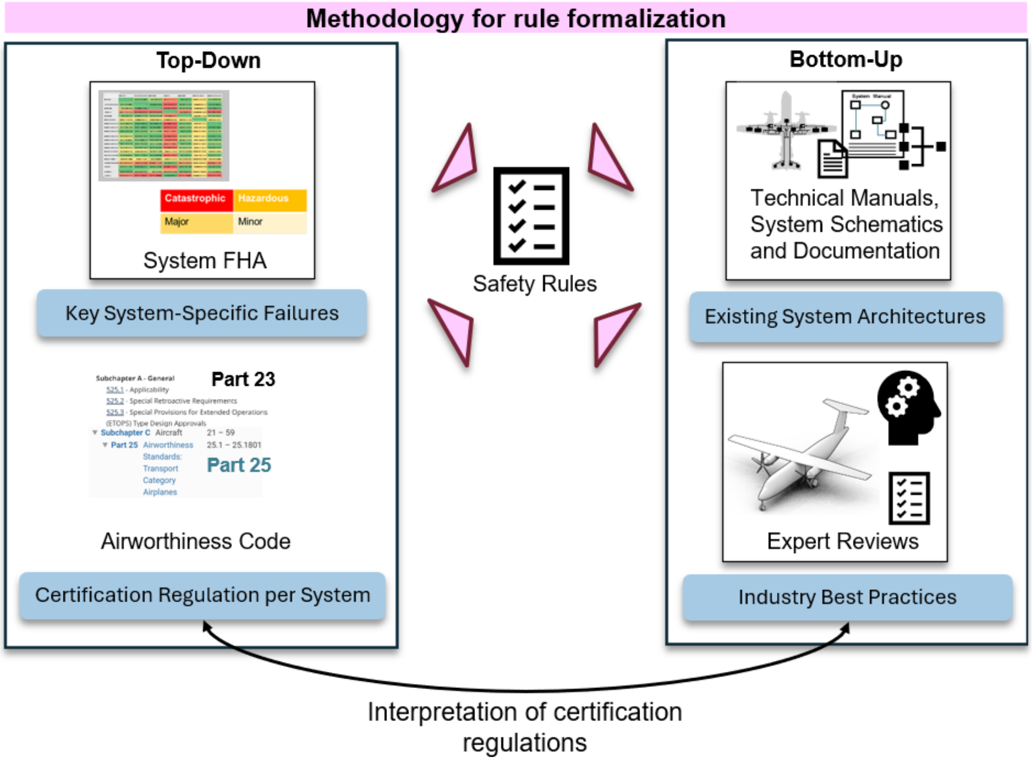

Methodology for rule formalization
Top-Down
Catastrophic
Hazardous
Major
Minor
System FHA
Key System-Specific Failures
Subchapter A - General
Part 23
25.1 - Applicability
25.2 - Special Retroactive Requirements
25.3 - Special Provisions for Extended Operations (ETOPS) Type Design Approvals
Subchapter C Aircraft 21 – 59
Part 25 Airworthiness Standards: Transport Category Airplanes 25.1 – 25.1801
Part 25
Airworthiness Code
Certification Regulation per System
Safety Rules
Bottom-Up
System Manual
Technical Manuals, System Schematics and Documentation
Existing System Architectures
Expert Reviews
Industry Best Practices
Interpretation of certification regulations

Figure 3.8: Methodology for safety rule identification and formalization

Bottom-up analysis requires analysis of the system architecture as documented in schematics and other system documentation. This can be cumbersome, as the documentation is detailed and the patterns in redundancy, power, and control allocation are described at a high level. To improve the clarity of understanding and to analyze the architecture, the generic element descriptor is used as an abstraction tool to resolve key information such as power allocation and redundancies and to identify power flows. Combining this information with other architecture information, such as the number of system components and the employed system technology, enables an initial identification of patterns in the specified redundancies of power systems and the logic in the allocation of power systems to system components. This process is repeated several times to consider different aspects of the system architecture - such as component technology and its influence on power allocation, the impact of technology and function criticality on the number of allocated system components for a specific function, and the allocation of power systems across different redundant system components – resulting in the synthesis of one or more safety rule statements.

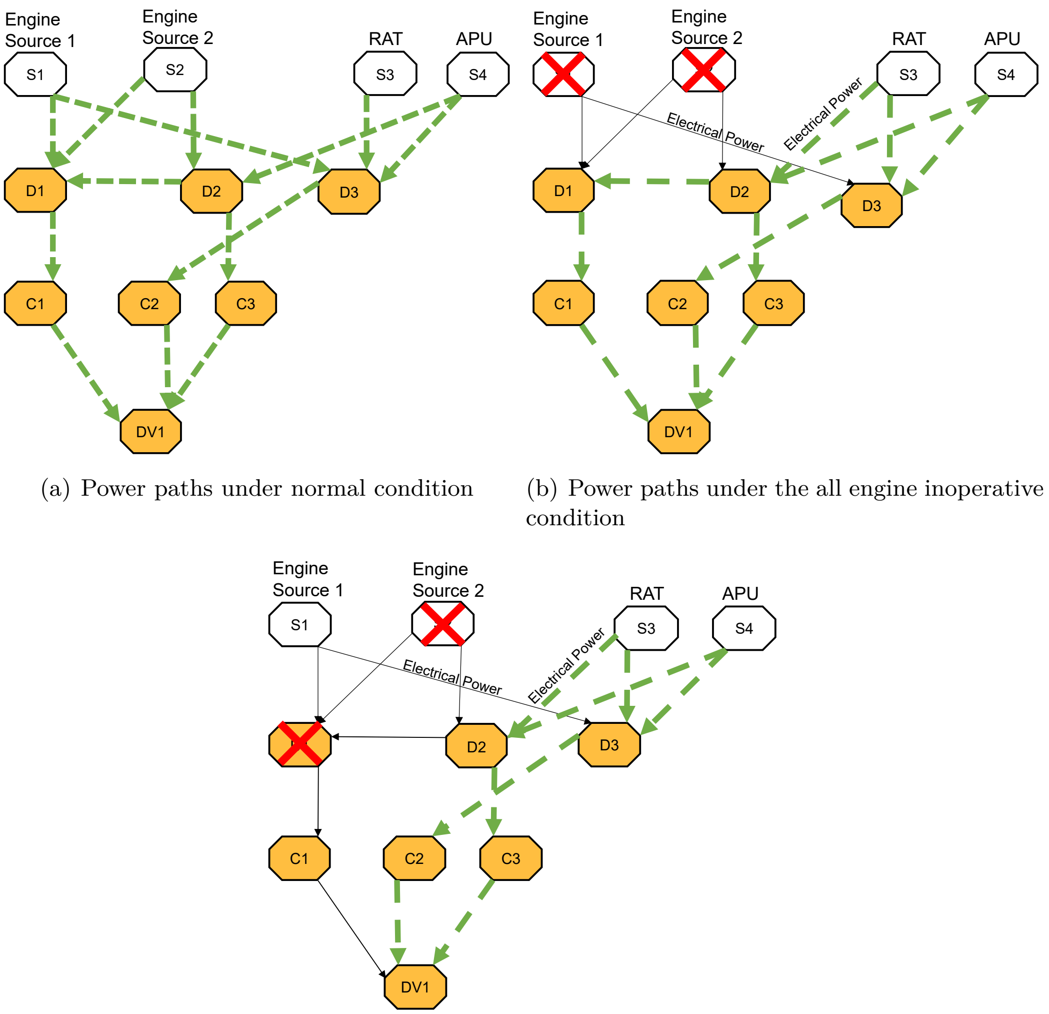
These safety rules are compared with knowledge-based engineering feedback in the form of design best practices and expert review. Points of ambiguity while defining safety rules, such as the type of components that can be considered a backup system or the general design philosophy behind specific system power allocations found in existing aircraft, are clarified and may result in the reformulation of the safety rule statement. Several rounds of reformulation, review and feedback are typically expected before a rule can be formalized.

The following sections will demonstrate the methodology for rule identification, generalization, and filtering. These are done using two aircraft systems: the landing gear braking system and the yaw control and actuation system. The latter represents a system with increasing complexity and a greater choice of technologies for actuation.

3.3.3 Rule Generalization and Extension to Unconventional/Novel Architectures

There are three categories of rules that are observed in practice while building and filtering system architectures. These are configuration-based, technology-based, and safety-based rules. Configuration-based rules focus on ensuring that a system architecture is sound in terms of its overall configuration; for example, in a landing gear braking system, the allocation of a brake actuation unit to a wheel on a different landing gear leg would be considered configurationally flawed. Technological rules focus on ensuring that the connections between components accurately represent the underlying technology; for example, a typical EHSA used for flight control actuation can only be supplied by one hydraulic system, whereas a hydraulic actuator with tandem cylinders is represented as a single component but can be supplied by two hydraulic distributions.

However, when existing system architectures are modelled using the generic element descriptor, an interesting characteristic of "power flow" emerges. Since the generic element-based descriptor is easily traceable, the flow of power from the source to different consumer elements becomes visually apparent. These can also be identified computationally as a simple path on a graph between two node sets [133,137]. Modelling system architecture from existing aircraft using the generic element descriptor enables the identification of typical patterns in power generation and distribution system redundancy characteristics. These can then be studied to define power-flow-based rules that stipulate the minimum number of power flows that each surface or device implementing a flight control function receives from the power sources. A rule base contingent on specifying a minimum number of power flows based on function criticality can be applied to both conventional and unconventional system architectures.



(a) Power paths under normal condition

(b) Power paths under the all engine inoperative condition

(c) Power paths with one engine and opposite distribution inoperative

Figure 3.9: Overview of rule generalization using the power path approach for the yaw control actuation and hydraulic power system of A320 aircraft

Consider figure 3.9(a), which shows the hydraulically powered rudder actuation system for the Airbus A320 using the generic element descriptor. Notice the power flows through each architecture originating from the source elements, going through the distribution and consumer elements, and being supplied to the device element. The power flows are detailed for three cases: a nominal case with all elements functioning, a failure case where both source elements have failed and finally a case where one source and the opposite distribution have failed. Under all conditions each device element is seen to have a minimum of at least two power flows from the source. Here, a power flow is considered an independent path without any regressive links from source to distribution in a directed graph representation of the system architecture using generic elements. Thus, a generic rule statement can be synthesized for hydraulically powered aircraft rudder actuation that states the following: Each device element representing an aircraft rudder must receive at least two incoming power flows.

To support the definition of the rule, an analysis of power flows on three different types of aircraft yaw control actuation systems is performed. In figure 3.9(b), the case where both engines have failed is shown, and it is evident that the rudder device receives two power paths starting from the remaining emergency sources, which include the RAT and the APU generator. If the APU generator is not considered, then one power path exists from the RAT to the rudder surface. In figure 3.9(c), the resulting power paths from another failure condition are shown wherein one engine and a hydraulic distribution associated with the other engine are lost, resulting in at least two power paths from the remaining power sources to the rudder even if the APU is not considered as an emergency power source. Similarly, variants of this failure case and other critical cases, such as a two out of three actuator failure, also result in a minimum of two power paths.

Thus, by invoking compliance by demonstrating similarity, it can be argued that any hydraulically powered yaw actuation system should have the same number of power paths under the same failure conditions. Therefore, for any surface or effector implementing a yaw control actuation function using hydraulic power, at least one power path is expected from source to device under the all-engine or power source failure condition, and a minimum of two paths are expected for the other two failure conditions. These are the single-engine/power source and opposite distribution failure conditions and two out of three actuator failure cases, and together, they are called the cardinal failure conditions in this thesis. A similar study of electrical yaw actuation and mixed actuation architectures, such as those on the Airbus A350 and Gulfstream G650, shows at least two hydraulic power paths and three electrical power paths to the rudder surface. Thus, the generic rules for yaw control actuation become the following:

- Rule 1: For hydraulically actuated yaw control effectors, a minimum of two power paths from source to actuation device are required

- Rule 2: For electrically actuated yaw control effectors, a minimum of three power paths from source to device are required

- Rule 3: For a mix of hydraulic and electric actuators, at least three power paths are required

The rules listed above can be applied in two ways that vary in the level of conservatism. The first approach is to ensure that the minimum required power paths terminate at each device element under nominal conditions with no failures. The second and more conservative approach is to ensure that the minimum number of power paths terminate at the device element under all cardinal failure conditions, leading to the configuration of the connections between backup sources (time-limited and non-time-limited), distribution and consumer elements. When using the first approach (for evaluative and generative filtering) a quantitative assessment of the system architecture is required to obtain a safety metric with which to quantify the available redundancy in power flow and compare different architectures that comply with safety rules.

3.3.4 Treatment of Control Elements

Section 3.3.1 introduced a set of generic elements to represent system architectures from a component redundancy and power flow perspective. However, in many aircraft systems, such as flight control systems, fuel systems, and aircraft braking systems, the metering of power is accomplished by electrical control using controllers or other electrically powered metering devices. Thus, from a safety perspective, the “control view,” which includes the allocation of controllers to actuators and the further allocation of power supplies to controllers, becomes important in evaluating the system architecture.

To support the representation of a control view of the architecture, the control element denoted by "Ctl" is added to the generic element catalogue and represents controllers or other power metering components. The control element receives electrical power and supplies a control signal to other elements. A typical example is shown in figure A.1 of section A.2, where actuators represented using consumer elements receive both power and control inputs. The control element provides the control input while itself being supplied with electrical power from an electrical distribution element.

Architecture representation using generic elements that include a control element can have logic associated with the control elements and the consumer elements to which the control elements are connected. The control element can be said to be in a working state if it receives a power input from a distribution element of the appropriate power type. Similarly, a consumer element representing an actuator works only when it receives both a control signal from the control element and power from a distribution element of the appropriate power type. This type of logic is useful for writing transfer functions when building safety models to enable automated fault tree generation and will be discussed further in section 3.6.2

3.3.5 Application: Rule Formalization for Aircraft Landing Gear Braking Systems

This section presents an example of rule identification for a landing gear braking system architecture. Powered landing gear wheel braking systems on aircraft typically consist of a powered actuation mechanism that applies pressure to a brake disc, allowing the motion of the disc to be arrested. This requires the appropriate type of power to be generated and supplied by typically redundant distribution systems. However, the aircraft’s certification basis can affect the rules that are applicable in the design of powered braking systems, and as such, the safety rules can differ according to different certification standards.

Certification rule analysis

First, using a top-down approach, an analysis of the 14 CFR Part 25 regulations drives the development of the safety rules. However, to show how the rules can vary based on regulations, the Part 23 regulations will also be described here. 14 CFR 25.1309 [125] and 23.2510 [138] are important certification rules that prescribe safety requirements that require the installed systems to function nominally under any foreseeable aircraft operating condition. These further stipulate that any failures leading to an unsafe condition of the aircraft must be extremely improbable.

Aircraft power systems have to be designed to supply essential loads under different operating conditions. Part 25.1310 defines essential loads as “Each installation whose functioning is required for type certification or under operating rules and that requires a power supply is an “essential load” on the power supply.” It further details that essential loads need to be supplied in cases of power source failure, such as prime movers and engines, or in situations where an essential load relies on an alternative source of power in case of the primary power source failure. The need for system redundancy in both the generation and distribution of electrical power is emphasized in 25.1355. In case of failure of an electrical power source, an alternate source along with a dedicated feeder or distribution is still required. Based on these certification regulations, it is evident that primary sources and alternate or backup sources of electric power are required, in addition to redundant distribution systems.

Analysis of failure conditions

The second aspect to consider is the key failure conditions that could affect the landing gear braking system. These are typically derived by performing a functional hazard assessment. Key failure conditions, which are characterized by the loss of common resource systems such as power generation and distribution systems, are identified using the FHA as being critical failures leading to the catastrophic loss of the braking function. It is important to note that the failure conditions listed here do not represent an exhaustive set but are selected because they have been identified as drivers of added redundancy and independence requirements. The failure conditions are the loss of all engines, the combined failure of one engine and the opposite distribution system and asymmetric braking. These failures are adapted from the FHAs performed for legacy aircraft programs employing hydraulic brake actuation systems and from guidance material [22, p. 383].

Analysis of existing systems architecture implementations (Bottom-up)

The third step in the rule identification process is to survey existing aircraft landing gear wheel braking systems from publicly available documentation, such as flight crew operating manuals or maintenance manuals. Consider the hydraulically powered landing gear braking system on the De Havilland Dash 8 aircraft. Hydraulic power is generated by dedicated engine-driven pumps on each engine and is distributed using two independent hydraulic distribution systems. These are termed normal and alternate hydraulic systems. Each braking unit is supplied with both normal and alternate systems, which can provide pressurized hydraulic fluid in case one system fails. In addition to these two systems, the braking system has accumulators within the hydraulic system that provide pressure to enable parking brake application on the ground when the main systems are not pressurized.

The graph descriptor is used to identify the sources of power and to demonstrate how each distribution is connected to individual sources. Analyzing the descriptor and the original schematic allows the identification of primary power sources, such as the engine-driven pumps, and backup sources, such as the Ram Air Turbine, Battery packs or their sources. There is some ambiguity about what can be considered a backup source. Therefore, industry best practices help inform the type of components that are typically used as backup power sources. In this case, the batteries are not considered a backup source since batteries are typically only expected to provide power for short durations. Furthermore, electric motor pumps in hydraulic systems cover transients in hydraulic flow demand. Taking all the aforementioned architecting and safety considerations into account, the following rules are identified that govern the configuration of landing gear wheel braking system architectures using hydraulic power, electric power, or a combination of both.

- Rule LDG - 1: Number of independent power supply sources. This rule ensures that at least two main hydraulic systems supply each hydraulic braking consumer. The backup system could be an independent power source or be supplied locally (using an accumulator or human-powered supply).
  - Rule LDG - 1a: For aircraft to be certified under Part 25, each braking consumer device must be supplied with at least two independent hydraulic sources and one backup source. In the case of electrical braking systems, at least three independent electrical power distribution systems must be provided

for each electrical braking consumer. This rule may also be satisfied by using an independent backup source connected to at least one of the electrical distribution systems, along with connections from other source elements. In this case, two independent electrical systems will suffice.

  - Rule LDG - 1b: For aircraft to be certified under Part 23, each braking unit must be supplied by at least a main hydraulic distribution and a backup distribution.

- Rule LDG - 2: Power supply must be allocated symmetrically to prevent asymmetric braking in case of power loss. Rule 2 focuses on preventing asymmetrical braking due to loss of specific power systems - this is a case that is typically identified during the Aircraft-Level Functional Hazard Assessment (AFHA).

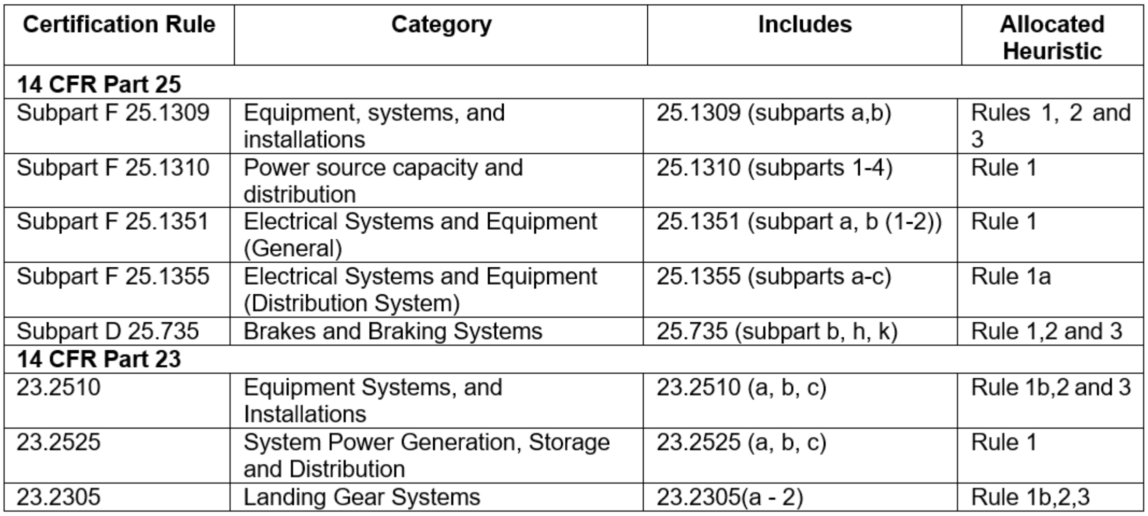
- Rule LDG - 3: At least one power-consuming braking device is allocated to each wheel. This rule filters unfeasible allocations between the braking device and the wheels for conventional landing gear braking systems. The rule-checking algorithm analyzes the generic element-based architecture descriptor and evaluates each rule individually.

- Rule LDG - 4 - Controller redundancy:

  - Rule LDG - 4a: For electrical braking system architectures, one brake control unit is to be allocated for every inboard-outboard wheel pairing.
  - Rule LDG - 4b: Two sub-controllers must be allocated to each wheel pair for every brake control unit.
  - Rule LDG - 4c: At least two electrical distribution elements must supply each controller.

Comparing these regulations with existing system architecture implementations allows ambiguities to be resolved and a generic heuristic to be developed. For instance, in the case of landing gear braking systems, it was identified that analysis of Part 25 certification basis results in two primary sources of power for conventional (hydraulic) braking systems. These are also supplied with a third backup source that fulfills a secondary braking function, such as parking. However, upon making a similar comparison for the Part 23 certification basis, it was noted that only a single primary source of hydraulic power was required, along with a backup source. Here, the main/primary, alternate, and backup distributions are characterized as being supplied by primary power generation sources (main and alternate) and secondary power sources (backup). These secondary sources can range from manual hydraulic actuation, such as an accumulator augmented hand pump, to an independent backup power source, such as an APU-driven pump or generator. Feedback from subject-matter experts and industry best practices indicated that RAT, APU, and non-time-limited independent power sources could be considered as backup power sources.

Table 3.2: Relation of proposed heuristics to aircraft certification basis



Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart F 25.1309	Equipment, systems, and installations	25.1309 (subparts a,b)	Rules 1, 2 and 3
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subparts 1-4)	Rule 1
Subpart F 25.1351	Electrical Systems and Equipment (General)	25.1351 (subpart a, b (1-2))	Rule 1
Subpart F 25.1355	Electrical Systems and Equipment (Distribution System)	25.1355 (subparts a-c)	Rule 1a
Subpart D 25.735	Brakes and Braking Systems	25.735 (subpart b, h, k)	Rule 1,2 and 3
14 CFR Part 23			
23.2510	Equipment Systems, and Installations	23.2510 (a, b, c)	Rule 1b,2 and 3
23.2525	System Power Generation, Storage and Distribution	23.2525 (a, b, c)	Rule 1
23.2305	Landing Gear Systems	23.2305(a - 2)	Rule 1b,2,3

Table 3.2 shows how the proposed heuristics are related to specific certification regulations - it is possible to consider that heuristics extracted from an analysis of certified aircraft (rules 1, 2, 3, & 4) are able to satisfy these regulations. Similarly, rule 1 also finds a basis in 25.1310, which stipulates the need for alternate power sources for essential loads. A similar finding is apparent in 23.2525. 25.1351 and 25.1355 deal with electrical distribution systems and are relevant to rule 1a in the prescription of three independent electrical systems for electric landing gear braking. Furthermore, 25.1351 references back to 25.1309 and thus provides a similar basis for rule 1a as that for rule 1 with 25.1310.

Rule evaluation

The generic element descriptor consists of its various elements linked together by connections that indicate power or control flows. The type of power each element uses and is supplied with is also captured in the descriptor. Rule evaluation is enabled by inspecting each node and checking if the appropriate type of power is supplied to it. Furthermore, the number of redundant components and power supplies can be checked by parsing through the list of source elements and through the chain of connections from each source element to associated distribution elements.

An example of rule assessment in L0 SysArc is shown in figure 3.10. The connections between sources (S1 & S2) and distribution (D1 & D2) in SysArch_XX7 ensure that two independent distribution systems always supply the consumer (C1). This is valid because S1 and S2 are independent primary sources.

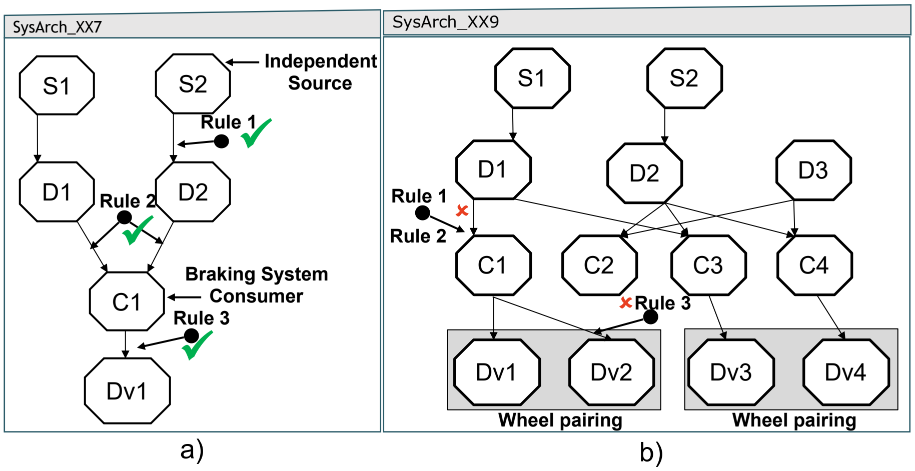


Figure 3.10: Methodology for safety rule identification and formalization

Additionally, C1 is allocated to a unique device, Dv1, and complies with rule three. However, SysArch_XX9, in figure 3.10 b), presents a case that fails the checks for compliance with rules 1, 2, and 3. Here, the consumer C1 is only supplied by one independent primary distribution, D1. This violates rule 2 as C1 is supplied asymmetrically to other braking elements (C2–C4). Finally, C1 is allocated to both Dv1 and Dv2, which is a violation of rule 3. Another approach to applying rule-based safety filtering is comparing the architectures in the design space to those built using a generative filtering approach.

Complex system: Rule formalization for aircraft yaw control actuation

The aircraft yaw control actuation system comprises components that enable the manoeuvring of the aircraft about the directional axis. For conventional aircraft, the primary yaw control surface is the rudder. Additionally, the yaw control actuation system features actuators that move the control surface in response to a pilot command. These can be unpowered, such as in the case of direct mechanical linkages from the pilot's control input to the actuation of control surfaces. They can also be powered, such as flight controls featuring actuators that are either powered hydraulically or electrically to move control surfaces. These actuators are associated with power distribution systems for electric or hydraulic power, which are typically duplicated or triplicated for redundancy. The level of redundancy of the actuator is determined based on several factors, such as the criticality of the flight control function, the actuation and power technology employed, and flight performance requirements.

Flight control actuators are differentiated on the basis of the type of power they use as well as the means by which they are controlled. Mechanical flight control actuation systems are both controlled and actuated by a system of mechanical linkages, pushrods, control quadrants, and pulleys. Mechanical controls are also found on hydraulically powered actuators, where a mechanical linkage actuates a control valve on the actuator. The introduction of fly-by-wire (FbW) systems enabled the electrical control of hydraulic actuators, where electrical signals are used to meter the flow of hydraulic fluid to the cylinder using a servo-valve. Electrically powered actuators such as the Electro Hydrostatic Actuator (EHA) or the Electro-Mechanical Actuator (EMA) are also controlled electrically.

The identification of safety rules for a single-rudder-based yaw control actuation system must consider the design space of different actuation and control technologies. Furthermore, an important aspect of safety is the redundancy requirements in actuator allocation and power supply. These can vary based on the actuator technology as well as the type of power source, such as primary or backup sources, that supply the actuator. The transition to More Electric Aircraft (MEA) system architectures has resulted in the incremental introduction of electric actuation systems in tandem with hydraulic actuators. Therefore, the allocation of power systems and the number of backup sources supplied to individual actuators can become complex when compared to conventional hydraulic systems.

The following section will focus on applying the steps required to identify safety rules, starting with a top-down analysis of certification regulation, followed by an analysis of critical failures of the yaw control actuation function using an FHA, and concluding by developing insights from an inspection of existing yaw control actuation system implementations according to the employed actuation technology.

3.3.6 Application: Aircraft Yaw Control Actuation System

This section presents an analysis of certification rules and typical failure conditions, as well as a summary of physical implementations pertaining to the design of aircraft rudder actuation systems fulfilling the yaw control function. It is important to note that the analysis of failure conditions is not exhaustive and only considers failures that have an impact on the level of redundancy required in the power generation, distribution and actuation systems to meet the minimum safety requirements for a critical function.

Certification rule analysis

The certification regulations pertaining to the configuration of a yaw control actuation system are focused on the three areas listed below:

1. Concerning the ability to meet minimum control requirements and compensate and overcome asymmetric yaw upon the loss of a single engine (for twin engine aircraft) - Part 25.147 [139] a) and b)

2. Concerning the ability to provide the yaw control function upon the loss of an engine and a power distribution system - Part 25. 1310 a) 1 - 4

3. Concerning the ability to meet minimum control requirements under a single failure or upon the loss of both engines (for a twin-engine aircraft) - Part 25.671 [140] c) 1-3, d)

The first point is interpreted to impact the number of actuators attached to a surface, as in the case of a single engine failure, where the actuation system should be able to provide sufficient rudder deflection to overcome the asymmetric yaw induced by the active engine. This means that for a given actuation and associated power system architecture, the number of actuators should be such that they must be able to provide the requisite hinge moment to deflect the surface enough to provide a heading change of up to fifteen degrees in the direction of the critical inoperative engine. Thus, a sufficient level of redundancy in the number of actuators coupled with the sizing of the actuators to these critical conditions is required. An example of the required level of redundancy to meet these requirements would be an aircraft with three actuators on the rudder surface and being able to meet the one engine inoperative requirements at the minimum control speed (as defined by Part 25.149 [141]) with two of the three actuators failed.

The third item on the list of certification regulations shown above is interpreted as specifying the need to have an emergency power system independent of engine-based secondary power generation with which to power the rudder actuators in the event of an all-engine failure. This is typically accomplished by using a RAT on large aircraft.

Analysis of failure conditions

Based on consultation with industry experts, typical failure conditions to consider when analyzing rudder actuation system architectures include, but are not limited to, the following:

1. Loss of engine alongside loss of yaw trim function
2. Inadvertent activation of thrust asymmetry compensation commands
3. Reduction of yaw control authority below minimum control requirements

The first point is interpreted as a need to ensure the reliability of the power system, the actuators, and the allocation of control and power to the actuators, such that the inability of the actuation system alone to meet MCR upon the loss of an engine is unlikely, without the use of trim functionality.

The second point deals with uncommanded activation of thrust asymmetry compensation and could have implications in the design of the actuation control system and the allocation of control elements to the actuators.

The third point has implications for both the power system and the actuators. The loss of an independent power supply system and of actuators could result in insufficient yaw control authority to meet the Minimum Control Requirements (MCR). Thus, one of several mitigation strategies to be applied is to ensure that the required redundancies in the power system, in actuator allocation, and in the allocation of emergency power sources to actuators are sufficient.

Rule formalization

Based on the analyses outlined above and the analysis of existing yaw control actuation architectures presented in appendix A.1, the following rules are specified for configuring the power and actuator allocation for yaw control using a single rudder surface for hydraulic and more-electric actuation technologies.

Hydraulic actuation

- Rule HFCA[1]- 1: The rudder surface shall have at least three EHSAs
- Rule HFCA - 2: The rudder EHSAs shall be provided by three independent hydraulic systems - one allocated to each actuator
- Rule HFCA - 3: One hydraulic system must be supplied by an independent emergency and non-time-limited power source

Electrical actuation

- Rule EFCA[2] - 1: The rudder surface shall have at least three EHAs or EBHAs
- Rule EFCA - 2: The rudder EHAs shall be provided by two main and one emergency electrical system

[1] Hydraulic Flight Control Actuation

[2] Electrical Flight Control Actuation

- Rule EFCA - 3: At least one EHA should be supplied with power from the emergency electrical bus and should be sized to meet MCR
- Rule EFCA - 4: At least two independent emergency power sources need to be present, and one of these sources should be non-time-limited
- Rule EFCA - 5: If EBHAs are used, there must be at least two hydraulic systems to supply three EBHAs
- Rule EFCA - 6: One hydraulic system must be supplied by an independent emergency and non-time-limited power source

Mixed electrical and hydraulic Actuation

- Rule MFCA[3] - 1: At least three actuators should be allocated to the rudder surface
- Rule MFCA - 2: If EHSAs and EBHAs or EHSAs and EHAs are used together on a rudder surface, two independent hydraulic and three electrical systems must be provided. At least one hydraulic system and one electrical system must be connected to an emergency power source

A summary of the aforementioned safety rules is provided in section A.3 of appendix A.

3.3.7 Application: Rule Formalization for Fly-by-Wire Flight Control Actuation Systems

The system architecture of FbW systems must enable proper execution of system functions under a variety of failure conditions. From an aircraft conceptual design point of view, the choice of control architecture and the allocation of control elements to different actuation elements can influence the overall weight of the system. A typical example is the additional wiring required for centralized Actuator Control Electronics (ACE) versus distributed Remote Electronic Unit (REU), sometimes known as Remote Actuation Unit (RAU). Centralized actuator control electronics require wires to be run from each ACE to the individual actuator it controls when compared to remote electronic units, which only need to be connected to a digital bus and can result in weight savings on a large aircraft [142].

FbW systems replace mechanical signalling with electrical signalling of flight control actuators. This typically requires equipment to receive the pilot's control input through a yoke or side-stick. Flight Control Computers (FCC) process the input and generate signals to command the movement of the actuators that will yield the required control surface deflection to meet the pilot's command. In addition to these elements, the FCCs also interface with multiple air data sensors and send commands to the actuators through ACEs and REUs. FCCs also enable envelope protection and flight control augmentation functions and are programmed to operate under different flight control laws.

The FbW system switches through the different FCLs upon encountering FCC failures, electrical or hydraulic system failures, and sensor failures, among others. FbW system operation under normal law typically includes all the expected flight envelope protections and augmentation functionalities of the FCC. Alternate law is activated when there are flight control computer failures or hydraulic or electrical system failures, resulting in reduced functionality. Finally, if there are additional failures in the aircraft power system or the

[3] Mixed Flight Control Actuation

FbW system, then direct law is activated and involves the pilot inputs being directly sent to the actuators through the ACE or REUs without going through the FCC. Thus, the allocation of ACEs and REUs to aircraft primary flight control surface actuators also has a significant bearing on the safety of the FbW system.

The following sections will detail the identification of safety rules for the specification of FbW architecture from the perspective of the allocation of control elements to primary flight control actuators. Here, the main assumption is that the operation of the FbW system under direct law occurs through the actuator control elements and represents a critical scenario beyond which further system function degradation cannot be tolerated.

Analysis of certification regulations

This section focuses on examining relevant certification regulation that influences the specification of FbW system architectures.

Aircraft flight control systems for transport category aircraft are required to be compliant with the FAR Part 25 regulations such as 25.1309 and 25.671 [140]. Part 25.671 (c) and (d) specifically focus on ensuring the control system enables the aircraft to be capable of continued safe flight and landing without exceptional piloting skill or strength in the event of certain failures or combinations of failures. These include any single failure, such as failure of mechanical elements, or structural failure of hydraulic elements, such as hydraulic actuators; combinations of failures that are not shown to be extremely improbable, such as dual hydraulic or electrical failures; or a combination of single failures with any probable hydraulic or electrical failures [140]. Part 25.671 (d) specifies that the aircraft must be controllable if both engines fail.

The certification regulations guide the focus of control system design to ensure that the aircraft can still be reasonably controlled and manoeuvred to landing without exceptional piloting effort. The various failures that Part 671 (c) and (d) specify directly influence system redundancy for each control axis, and Part 671 (d) focuses on ensuring control system operation in the event that engine power sources are no longer available due to an all-engine failure.

Thus, the allocation of ACEs and REUs to control surface actuators and the allocation of power to these controllers need to ensure operation in the event of multiple power systems and all-engine failures. This results in the need to ensure that the controllers are supplied from both primary (engine-based) and emergency power supplies. Furthermore, part 671 (d) could potentially imply the need to have a control path that is dissimilar from and independent of engine-based power.

Based on the above-presented analyses and an evaluation of existing FbW architectures shown in appendix A.2, the following patterns in the configuration of FbW architectures are identified:

1. Centralized architecture with three primary computers, three secondary computers, an analog ultimate backup controller, and no dedicated remote electronic units

2. Centralized architecture with three primary flight control computers, dedicated ACEs (typically four) for actuator control loop closure, and an ultimate analog backup controller signalling requisite actuators per control surface to fulfill minimum control requirements

3. Distributed architecture with three primary flight control computers, REUs distributed across surfaces, and an analog ultimate backup connected to as many actuators per surface as is required to meet minimum control requirements

4. Centralized architecture with three flight control computers (typically dual channel), four actuator control units, and remote units on the actuator for digital-to-analog conversion

5. Distributed architecture with three primary flight control computers and two types of ACEs with each ACE connected to a single actuator such that in the event of the loss of all actuators of a certain type, enough actuators can still be signalled using the remaining ACEs so as to meet minimum control requirements

The choice of controller configurations can be influenced by a number of factors, such as supplier choice, cost, weight, system-level requirements, and the confidence of the certification authority in the ability of the system supplier and integrator to provide a safe solution. For example, option five is typically implemented using single-channel REUs, while option three is implemented using dual-channel REUs. Anecdotal evidence suggests that on a recent large business aircraft program a variant of option three without the analog backup and with two dissimilar types of REUs was not selected due to cost and concerns expressed by the certification agency. To help the system architect configure a new FbW control architecture with the minimum required component redundancies, control type dissimilarity and redundant power supply allocation that are present in certified FbW implementations, the following rules should be followed:

Primary Flight Control Computer

- Rule FCC - 1: Three primary FCCs shall be present in a FbW implementation
- Rule FCC - 2: Each computer shall have dissimilar hardware and software
- Rule FCC - 3: Each computer shall have a minimum of two channels
- Rule FCC - 4: Each computer shall be supplied with at least one independent emergency source in addition to independent main sources
- Rule FCC - 5: Each computer shall be physically segregated to mitigate against the common mode failures captured under the PRA
- Rule FCC - 6: A dissimilar analog control unit shall be made available and connected to a sufficient number of actuators to meet minimum control requirements
- Rule FCC - 7: The backup unit shall be supplied by an independent power source and an emergency power source
- Rule FCC - 8: Only non-time-limited sources will be considered as an emergency source[4]

Remote Electronic Units

The following rules are based on the assumption that dual-channel REUs will be used in the FbW control architecture, so that each REU is capable of controlling two actuators.

- Rule REU - 1: For conventional aircraft, 2 REUs shall be prescribed for each surface except those implementing yaw control

[4] Time-limited sources can be specified in later design stages as a means to provide power while non-time-limited emergency sources are being deployed or when main sources are being restored.

- Rule REU - 2: For those surfaces implementing yaw control, three REUs shall be prescribed [5]

- Rule REU - 3: REUs must be powered by electrical sources independent of the aircraft's main electrical systems (e.g. using PMGs)

- Rule REU - 4: At least two REUs per surface must also be powered by emergency electrical power sources

- Rule REU - 5: REUs must be locally segregated according to the outcome of the PRA

- Rule REU - 6: Using the control view, each REU must be the terminal element in at least two power paths from source to consumer element (REU)

Rule generalization for novel aircraft

The above-mentioned rules are derived from an analysis of conventional aircraft, which feature a familiar tube-wing configuration. However, novel aircraft may present multiple surfaces implementing a flight control function, which can also be spread across different axes. As such, the rules are generalized and can be applied to any aircraft FbW control architecture.

Primary Flight Control Computer

- Rule FCC - G[6] - 1: Three primary FCCs shall be present in an FbW implementation

- Rule FCC - G - 2: A dissimilar ultimate backup controller must be present in an FbW implementation

- Rule FCC - G - 3: The ultimate backup must be connected to a minimum number of components implementing the flight control function on each axis such that minimum control requirements are met

- Rule FCC - G - 4: Power allocation is as outlined in the list above (3.3.7)

Remote Electronic Units

- Rule REU - G - 1: A minimum of two REUs are required to control a component implementing a flight control function along each flight control axis

- Rule REU - G - 2: Three REUs are to be applied to a component if it fulfills more than one critical function, including flight control (rotors for lift and for pitch)

- Rule REU - G - 3: At least one REU on each component implementing a flight control function must be supplied with an emergency source in addition to a main source

- Rule REU - G - 4: A component implementing multiple functions, including flight control, must have at least two REUs supplied by main and emergency sources, and the main source must be different for each REU

- Rule REU - G - 5: Segregation requirements are according to the REU configuration rules listed above

[5] If yaw control is implemented using multiple surfaces, two REUs may be assigned per surface.

[6] Generalized

One type of novel aircraft is the electric Vertical Take-off and Landing (e-VTOL), which is an aircraft poised for applications in urban air mobility. Although reliable data is not available on the control systems of these aircraft, it is to be noted that longstanding FbW system suppliers such as Thales have begun to market solutions specifically for e-VTOL and electric aircraft [143] that follow the three PFC and ultimate backup approach used in modern FbW aircraft and captured in the rules above.

3.4 Integrating ASSESS-L0 with MDAO Workflow

This section discusses the information transfer from the graph descriptor and the system architecture specification in an MBSE environment to multidisciplinary analysis and optimization (MDAO) workflows.

Integrating safety assessment within MDAO requires information about the system architecture at sufficient levels of detail for both safety assessment methods and system sizing methods. Each analysis requires different levels of detail and different types of information. For example, an analysis of system architecture safety requires information about interconnections between different system elements, detailing of power and control flows and, in some cases, component-level data. In contrast, a system sizing analysis integrated within an MDAO workflow typically requires a low-level to mid-level fidelity description of the architecture and sizing parameters. A few examples of the latter are the hydraulic system pressure, the flow rate, and the control surface hinge moment for the sizing of hydraulic and flight control actuation systems.

Since system architecture information is typically stored in a system architecture descriptor, any integration of safety assessment with an MDAO workflow (containing system sizing tools) will require a descriptor from which information necessary for both domains can be derived. At the same time, considering that the scope of the safety assessment is limited to conceptual design, the system architecture descriptor must be human-readable and at a higher level of abstraction than typical system architecture specification artifacts. Based on the level of detail, this section develops a link between two types of system architecture descriptors with multiple types of MDAO workflows. The first is the generic element descriptor, introduced in section 3.3.1, which supports early safety assessment using an abstracted and graph-based representation of the system architecture. The second type of descriptor features a higher level of detail than the graph-based descriptor and is a formal MBSE specification of the system architecture. Linking an MBSE specification with an MDAO workflow will enable the MBSE model to be enriched with the results of system sizing and architecture evaluation methods. Furthermore, the MBSE specification supports safety assessment activities, as will be shown in section 3.6.1.

The purpose of this link between the system architecture descriptor and the MDAO workflow is to enable architecture evaluation and to obtain aircraft-level metrics such as weight, fuel burn, and cost. Here, the focus will be on detailing modes of information transfer between the graph descriptor and the workflow based on the specific characteristics of the MDAO workflow. It is therefore essential to distinguish the different types of MDAO workflows, how each is developed in terms of tool integration, and the mode in which data is handled within each type of MDAO workflow.

According to van Ghent [144] an MDAO-based development process has two phases: the problem formulation phase and the problem execution phase. Problem formulation consists of developing an MDAO solution strategy that involves the composition of an

MDAO workflow specification which is a result of consolidating disciplinary tools, specifying tool connections, and selecting an MDAO architecture. MDAO workflow composition methods are typically categorized based on whether data from each tool is handled within the workflow as centralized or decentralized.

Decentralized approaches to handling tool data allow each tool the freedom to define its own local and global variables at the expense of the additional effort required to organize the interface between different tools and their global variables. Conversely, a centralized approach to data handling uses a formal nomenclature for variable names across the different tools in the workflow. This requires ensuring consistency in variable names across different tools, with the benefit of removing the need to link variables together across tools. A variant of the centralized approach is in the use of a Central Data Schema which consolidates different variables in a formal specification. An example of this is the CPACS schema developed by Nagel et al. [145], which prescribes a standard set of aircraft design parameters for use in aircraft design activities. CPACS can also be extended to incorporate custom parameters of different tools using the tool-specific section of the schema.

The interface with the MDAO workflows developed in this thesis deals predominantly with workflows using centralized data handling and central data schema. The focus is on providing system architecting data from the generic element descriptor to the inputs of the recipient system architecting tool within the workflow. The scope of the integration methods discussed here lies in managing the data transfer from the graph descriptor to three different types of workflows, each with different levels of granularity in their system architecture description inputs. These MDAO workflows are as follows:

- Distributed and Collaborative MDAO Workflow
- Active Industrial Workflow
- Notional Sandbox Workflow

The distributed and collaborative workflow is one in which the tool repository and tool owners are disparate, either organizationally or geographically, and where the tools are associated with different organizations or subgroups within an organization. The collaborative MDAO workflow used here is the AGILE 4.0 MDAO workflow created by the collaborating organizations in the AGILE 4.0 project. It features a centralized workflow development platform called KE-Chain [146] that integrates the input and output schema of each individual tool and combines it into a single schema that is exchanged between the tools during runtime. The workflow is integrated in RCE, which is a Process Integration and Design Optimization tool developed by the DLR; the native CPACS schema is utilized with several tools organizing their inputs and outputs using the tool-specific category within the schema.

The industrial workflow in question is part of the multi-level MDAO framework in use at Bombardier. This workflow is part of the Bombardier Engineering System (BES) and, as described by Piperni et al. [79], consists of Conceptual MDO (CMDO), Preliminary MDO (PMDO) and Detail MDO (DMDO), with each level addressing different aspects and employing tools and analyses of increasing fidelity. The CMDO workflow is focused on business case assessment, systems integration, design space exploration, and the selection of promising aircraft configurations, whereas the PMDO and DMDO focus on higher fidelity disciplinary analyses, such as aerodynamics and structural analysis, to refine and validate the configuration selected using the CMDO process. The entire workflow is implemented in the commercially available iSight platform [147].

The scope of this work is limited to transferring system architecture information to the systems tools in the CMDO workflow. The CMDO workflow has a systems integration discipline that itself comprises a nested workflow of systems sizing tools. The systems workflow is the specific focus of information transfer activities. The CMDO workflow stores the data exchanged between the tools as global variables in a text-based data schema.

However, the systems workflow uses a spreadsheet-based architecture descriptor that also includes system sizing inputs. The architecture is configured in the spreadsheet and the sizing outputs are written to the global text-based data schema upon execution of the systems sizing workflow. Thus, to provide a system architecture input to the systems workflow from the generic element descriptor, the information in the descriptor needs to be mapped to the architecture inputs in the spreadsheet-based descriptor.

Finally, the sandbox environment is a custom workflow developed in the aircraft systems laboratory at Concordia University. The workflow was first conceived as an environment to test and integrate new disciplines such as safety, certification, thermal analysis, and maintainability into an MDAO environment within the MDAO – NextGen project [28]. The objective was to use the sandbox environment to support the transition of new capabilities into the aforementioned industrial workflow. The sandbox workflow is described in detail by Sanchez et al. [84] and has since had several refinements. It consists of aircraft sizing, systems sizing, safety, and thermal risk as the main disciplines. It is also supported by the integration of a geometric CAD modeller to enable system placement in conceptual design and to enable maintainability analysis.

The system sizing tool ASSET is described in Mohan et al. [66, 148]. It categorizes aircraft systems into power generation, power distribution, and power consumption systems. It uses a mix of physics-based models and semi-empirical models to estimate the power consumption, weight, and fuel-burn penalty of aircraft system architectures. ASSET allows the configuration of the system architecture to include important system-level technologies and allocation of power supply to power-consuming systems components. The system architecture input is provided to ASSET using an XML-based data schema that classifies the information per subsystem and provides a section where the sizing outputs can be stored.

The workflows introduced in this section are synthesized for specific applications and, as such, differ in complexity and implementation. However, one element that they have in common is that the interface to the system sizing tool is through the system architecture descriptor. All the workflows use a system architecture descriptor, albeit at different levels of complexity, detail, and types of implementation. The descriptors used here can be classified as follows:

1. Standardized Data Schema using Mixed Fidelity Description of Systems Architecture
2. Unstructured Data Schema
3. Standardized Data Schema Using Rich Description of System Architecture

A rich description of the system architecture ensures that the resulting schema or data model is human-comprehensible, is intuitive, and is structured to be programmatically efficient for the retrieval and storage of data. This work defines the characteristics of a rich data model describing aircraft system architecture as being one where each subsystem and its parameters are clearly identifiable and differentiated with unique identification. Furthermore, the parameters of each subsystem are provided with unique names and can be accessed by specific keys. An important aspect in the system architecture description schema and resulting data model is that the allocation of power distribution system components to power consumption system components must be clearly defined as a parent-child relationship to improve readability of the schema and ease of parsing the data model.

Since the system architecture definition can often contain nested relationships up to several degrees, such as the allocation of an actuator to one or more power systems, the allocation of power systems to one or more generators, and the allocation of a generator to an engine. A rich description schema must allow the definition of the architecture with clear delineation of assignment and ownership relations. Additional considerations are the readability, modifiability, and variability of the description schema. This means that the schema should allow for the representation of unconventional architectures or atypical allocations, or at least be extensible to support new features.

Unstructured data schema are those in which the data is captured en-masse and where the architecture of the system may not be apparent. The system architecture data may be mixed with aircraft-level inputs, and there may not be a clear specification of the relationship between the data. Furthermore, codified flags may be used to instantiate specific configurations of the system architectures. For example, the data schema may contain a variable whose value, when set to 1, could indicate to the sizing tool that a specific hard-coded configuration of the system architecture is being used, which can trigger the use of specific sizing equations or semi-empirical models. Finally, mixed fidelity models are considered to be those that feature some aspects of the rich descriptors but do not entirely comply with the need to represent interaction, allocation and ownership between system architecture components.

MDAO integration concept workflow

This thesis proposes the integration of safety assessment into MDAO workflows through the system architecture descriptor and the use of the information contained therein to support safety analyses. Figure 3.11 shows a concept MDAO workflow with different disciplinary tools; the safety assessment tool is highlighted. The safety assessment tool consists of the ASSESS L0 and ASSESS L1-M1 modules, which take the system architecture information and perform safety analyses. ASSESS L0 uses information either directly from the generic element descriptor or, if another system architecture schema is used, converts it into the generic element descriptor. ASSESS L0 checks the safety rules and can perform basic quantitative analysis shown in section 3.6.2 to develop a safety metric for each architecture. ASSESS L0 can also be programmed to reconfigure the system architecture descriptor to add the minimum required redundancies in power flows if required. The ASSESS L1-M1 tool is envisioned to provide more detailed safety analysis for specific failure conditions that can be predefined by the architect.

The system architecture descriptor then provides the relevant information to the system sizing tool to determine the weight and power requirements of the architecture, which are further propagated towards the calculation of aircraft-level metrics.

MDAO integration studies

The key enabler for MDAO integration is the system architecture descriptor and its ability to store information. In the proposed framework, the system architecture descriptor provides a direct input to the MDAO workflow through the system sizing tool. This section first describes how the generic element descriptor can be linked to industrial and sandbox workflow, and then section 3.4.2 shows how the integration of the descriptor can

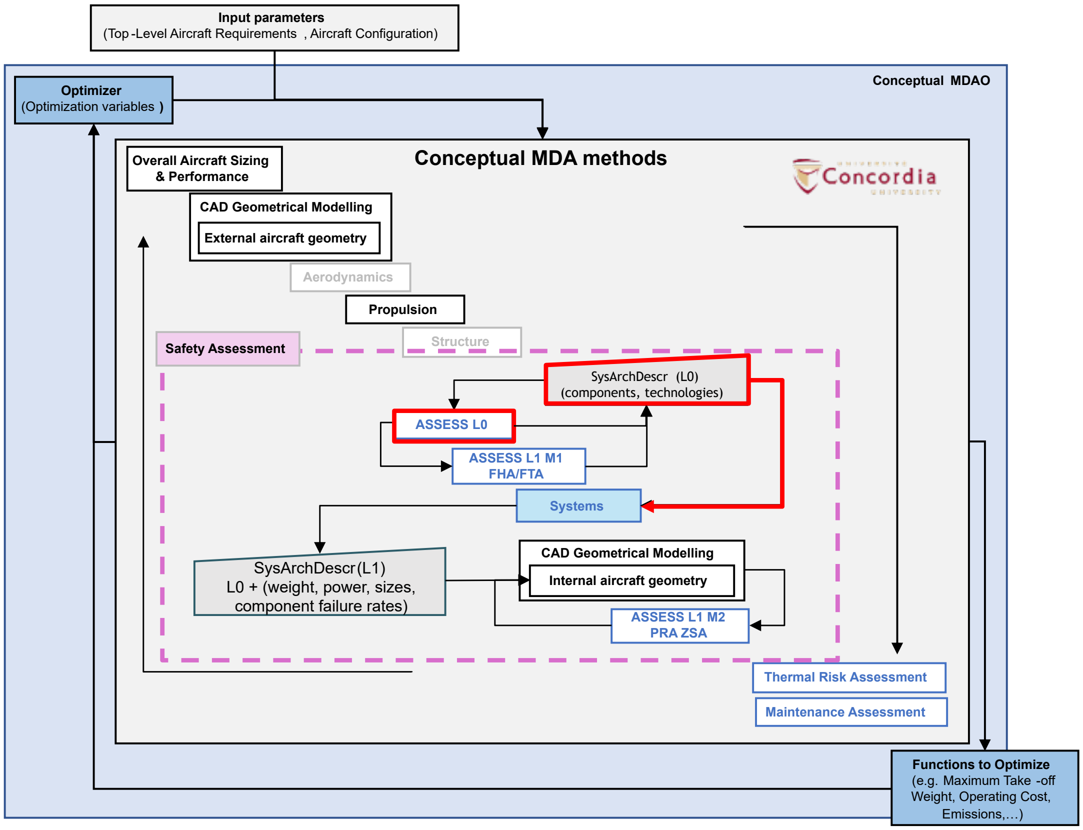

Input parameters
(Top-Level Aircraft Requirements , Aircraft Configuration)
Conceptual MDAO
Optimizer
(Optimization variables)
Conceptual MDA methods
Overall Aircraft Sizing & Performance
CAD Geometrical Modelling
External aircraft geometry
Aerodynamics
Propulsion
Structure
Safety Assessment
SysArchDescr (L0)
(components, technologies)
ASSESS L0
ASSESS L1 M1
FHA/FTA
Systems
SysArchDescr(L1)
L0 + (weight, power, sizes, component failure rates)
CAD Geometrical Modelling
Internal aircraft geometry
ASSESS L1 M2
PRA ZSA
Thermal Risk Assessment
Maintenance Assessment
Functions to Optimize
(e.g. Maximum Take-off Weight, Operating Cost, Emissions,...)

Figure 3.11: Concept workflow for the integration of safety assessment into MDAO workflows

enable safety assessment using the evaluation of safety rules for required redundancy. Two applications are studied: the first is the integration of safety assessment into a collaborative workflow using a CPACS-based system architecture descriptor, and the second is an industrial workflow using a spreadsheet-based system architecture descriptor.

3.4.1 Integration of the Generic Element Descriptor with an MDAO Workflow

This section describes the mapping between elements of the generic element descriptor and the input schema of both the industrial and sandbox workflows. The scope of this section is predicated on the introduction of the generic element descriptor as an enabler for evaluating safety rules and performing early quantitative safety assessment in conceptual design. However, safety metrics are but one of several indicators of the performance of a particular system architecture at the aircraft level. The others are the impact of the system architecture on overall aircraft weight, fuel burn, drag, etc. Thus, enabling safety to be included as a discipline in an MDAO workflow required the relevant information from the graph descriptor to be provided to the workflow as input to the safety discipline while providing relevant information about the architecture to other disciplines at the same time. Therefore, the scope of this section is limited to describing a mapping between the generic element descriptor and typical system architecture inputs that are provided to the systems sizing tool in an MDAO workflow. This section does not discuss the formulation of

MDAO problems dealing with system architecture but focuses on developing an information mapping under the assumption that if the information from the graph is provided to the workflow as the system sizing inputs in the expected format, the architect will then be able to evaluate the system architectures overall weight and performance impact in a standalone manner. Finally, since the graph-based descriptor can be evaluated for safety rules and can be used to derive quantitative metrics, the mapping described in this section allows safety assessment to be integrated as a discipline within the MDAO workflow.

Industrial Workflow

The scope of the industrial workflow discussed in this section is conceptual design, and it consists of tools that fall under the category of aerodynamics, aircraft sizing, performance, and systems, among others. The systems discipline itself consists of a systems workflow that receives aircraft-level inputs from the main workflow and also computes its own set of inputs from a user-supplied description of the system architecture. This system architecture descriptor is spreadsheet based and contains inputs that are categorized according to each aircraft system. In addition to system-related inputs, the descriptor also contains general sizing inputs specific to the tools within the systems workflow. Interfacing the graph-based architecture description involves writing the system architecture information to the spreadsheet-based descriptor.

In order to ensure that inputs are written in a traceable manner, the spreadsheet descriptor is modified to contain additional categorization. The proposed classifiers to structure the data in the descriptor are: "category," "component," "parameter," and "value". Thus, each category corresponds to a specific group of inputs, which can be either specific to a system or belong to a class of tool inputs. Within each category, there are components which represent either sub-components of a particular system, instances of a particular component, or general tool-related elements. Each component can have several parameters, which further will own a specific value, a string representing a unit, and another string which holds a description of that parameter.

The graph-based system architecture descriptor has some explicitly defined parameters associated with the different nodes; other input parameters must be inferred from the structure of the graph. For example, for an architecture with three hydraulic distribution systems represented using three "Distribution" elements, the pressure of each hydraulic system is stored as a property of each "Distribution" node in the graph, but the number of distribution systems can be inferred from the structure of the graph by identifying the number of nodes of type: "Distribution" and power type: "Hydraulic".

The information garnered from the descriptor must then be written to specific cells within the spreadsheet-based descriptor. The challenge with the implementation is the lack of dynamic modification of the spreadsheet-based descriptor, which is configured for traditional aircraft configurations such as those with a single rudder surface and a set number of flaps. In contrast, the generic-element descriptor can represent novel architectures and reference elements that represent novel or atypical aircraft configurations and associated systems. The overall recommendation to ensure extensible system architecture description on the workflow side is to switch to a schematic or XML-based data structure, an excerpt of which is shown in figure A.3 of appendix A.4.

Graph to spreadsheet-based descriptor

This section demonstrates the link between the graph descriptor and the industrial workflow through the spreadsheet-based descriptor using an example of a hydraulically powered yaw control actuation system architecture. The transfer of information from the generic element descriptor representing an aircraft yaw control actuation architecture is demonstrated for five input fields of the spreadsheet-based descriptor, which will be filled with information from the graph-based descriptor. These fields are directly related to the components of the architecture and to the allocation of actuation and power supply elements.

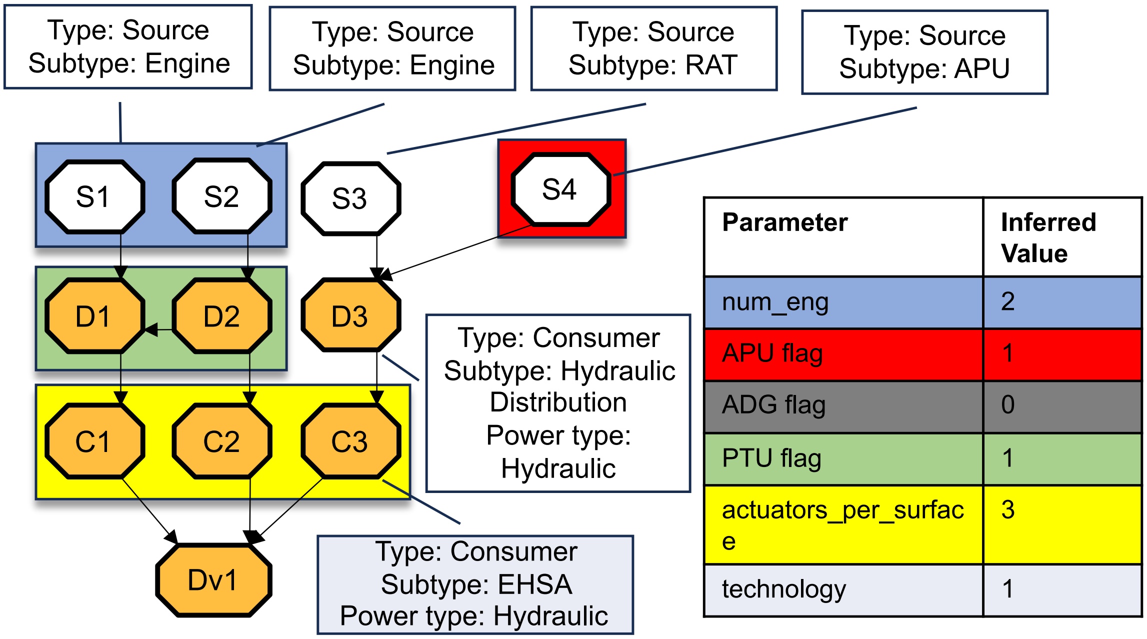
Consider the generic element descriptor shown in figure 3.12. It contains four source elements, three distribution elements, three consumer elements, and one device element. Sources S1 and S2 represent engine-based sources of hydraulic power. The distribution elements represent independent hydraulic distribution systems, and the consumer elements are hydraulic actuators. Each node stores properties that are defined by the system architect; for example, the source elements have a type property which identifies it as a source element, a subtype property that further defines the nature of the power source as being engine-based, and finally a power-type property that identifies that source as supplying hydraulic power. Similar properties are applied to the distribution and consumer elements.

The spreadsheet-based descriptor requires several input parameters to help specify the system architecture and to also specify certain key aircraft-level and system-level sizing parameters relevant to hydraulic and flight control system sizing. Some of these input variables and the parameters they represent are listed below.

1. num_eng: data type: int, number of engines
2. APU flag: data type: int - binary, on/off
3. ADG flag: data type: int - binary, on/off
4. PTU flag: data type: int - binary, on/off
5. actuators_per_surface: data type: int, number of actuators per surface
6. technology: data type: int - categorical, 1-EHSA, 2-EBHA 3-EHA

Parsing through the graph descriptor, the number of engines can be determined by iterating through the nodes of "type: Source" and counting the number of nodes that also have the "subtype: Engine" as shown in figure 3.12. Similarly, the Auxiliary Power Unit (APU) and Air Driven Generator (ADG) flags can be set by iterating through all the "type: Source" nodes and determining if any have the "subtype: APU". The "actuators-per" input can be ascertained by parsing through all the nodes of "type: Device" and counting the number of consumer nodes that are connected to each device. The "techn" parameter representing the actuator technology can be ascertained by reading the "subtype" property of each consumer node and allocating the correct value according to the listed actuation technology. The appropriate Power Transfer Unit (PTU) flag can be determined by parsing through the distribution nodes to determine if any two distribution nodes are connected to each other.

The link between the graph-based generic element descriptor and the spreadsheet-based descriptor can enable two-way information transfer with some important caveats. The spreadsheet-based descriptor can be fully populated with information from the graph-based descriptor, such that all relevant fields can be filled. Some fields may require information to be added to the graph-based descriptor, but the basic fields dealing with the system architecture, such as those listed above, can be easily ascertained from the descriptor. However, the reverse process of generating a generic element graph descriptor lacks the



Parameter	Inferred Value
num_eng	2
APU flag	1
ADG flag	0
PTU flag	1
actuators_per_surfac e	3
technology	1

Figure 3.12: Example of inferring inputs to the spreadsheet-based descriptor by parsing the graph descriptor

same level of completeness as the forward process. Information from the spreadsheet-based descriptor can be used to specify the number of source, distribution, consumer, and device elements, but it does not have enough detail to outline the connections between these elements. Here, safety and configurational rules can help by predefining connection possibilities. As an example, an actuator of "subtype: EHSA," represented by a consumer element, can only receive hydraulic power input from one hydraulic distribution system. For yaw actuation with a single rudder, all three actuators need to be connected to the same surface, and thus, the connections from consumer to device can be constrained.

Although there is less flexibility in defining the generic element descriptor from the spreadsheet-based descriptor, the transfer of information from the spreadsheet descriptor and its manifestation in graph format allows for two interesting possibilities. These are as follows:

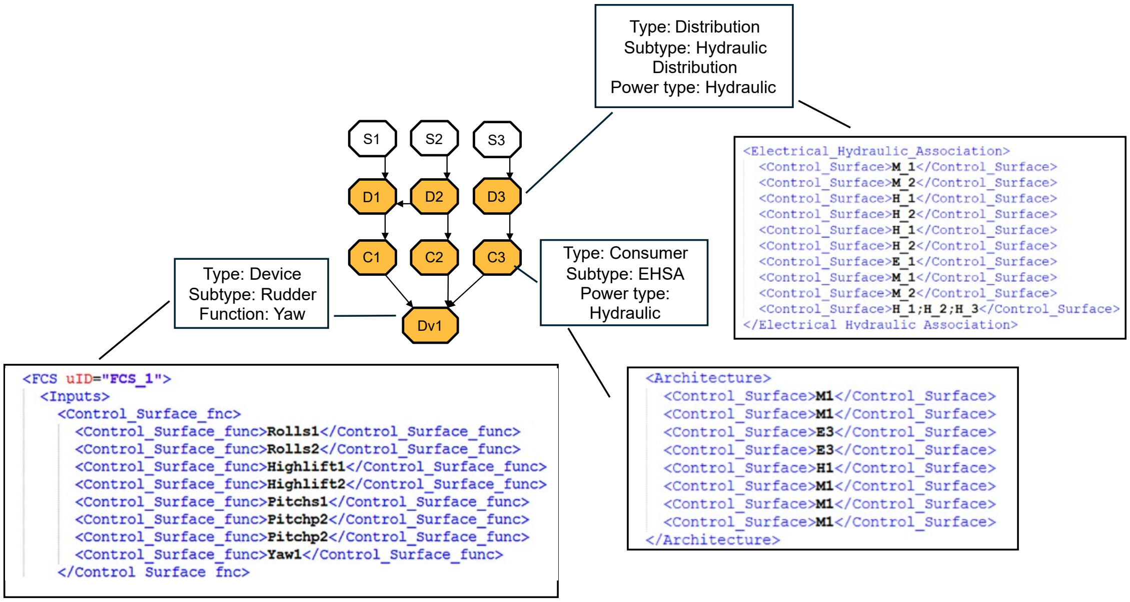
1. Interactive architecting to modify and manually define connections between elements or nodes of the descriptor

2. Automatically defining connections to explore the entire design space of resulting architecture options

In the case of the industrial workflow, the second point outlined above is demonstrated in section 3.4.2. Overall, the link between the graph-based system architecture descriptor and the spreadsheet-based workflow enables a system architect to either define a single system architecture and provide it to the MDAO workflow or provide a design space of system architectures that have undergone safety-based filtering and initial quantitative safety assessment to the MDAO workflow. Both options provide a means to integrate system architecture safety considerations into system architecture sizing and evaluation within an MDAO workflow.

Sandbox workflow

This section describes the mapping between elements of the generic element descriptor and the input schema of the sandbox workflow. The mapping enables system architecture inputs to be derived from the graph-based system architecture descriptor and adapted to the input schema of the systems sizing tool. This link enables the architect to evaluate the system architectures filtered in the system architecture definition and rule-based filtering step or to define custom architectures and evaluate the overall weight and performance impact in a standalone manner.



```xml
<Electrical_Hydraulic_Association>
  <Control_Surface>M_1</Control_Surface>
  <Control_Surface>M_2</Control_Surface>
  <Control_Surface>H_1</Control_Surface>
  <Control_Surface>H_2</Control_Surface>
  <Control_Surface>H_1</Control_Surface>
  <Control_Surface>H_2</Control_Surface>
  <Control_Surface>E_1</Control_Surface>
  <Control_Surface>M_1</Control_Surface>
  <Control_Surface>M_2</Control_Surface>
  <Control_Surface>H_1;H_2;H_3</Control_Surface>
</Electrical_Hydraulic_Association>
```

```xml
<FCS uID="FCS_1">
  <Inputs>
    <Control_Surface_fnc>
      <Control_Surface_func>Rolls1</Control_Surface_func>
      <Control_Surface_func>Rolls2</Control_Surface_func>
      <Control_Surface_func>Highlift1</Control_Surface_func>
      <Control_Surface_func>Highlift2</Control_Surface_func>
      <Control_Surface_func>Pitchs1</Control_Surface_func>
      <Control_Surface_func>Pitchp2</Control_Surface_func>
      <Control_Surface_func>Pitchp2</Control_Surface_func>
      <Control_Surface_func>Yaw1</Control_Surface_func>
    </Control_Surface_fnc>
```

```xml
<Architecture>
  <Control_Surface>M1</Control_Surface>
  <Control_Surface>M1</Control_Surface>
  <Control_Surface>E3</Control_Surface>
  <Control_Surface>E3</Control_Surface>
  <Control_Surface>H1</Control_Surface>
  <Control_Surface>M1</Control_Surface>
  <Control_Surface>M1</Control_Surface>
  <Control_Surface>M1</Control_Surface>
</Architecture>
```

Figure 3.13: Example of inferring inputs to the sandbox workflow by parsing the graph descriptor

The input schema to the system sizing tool is defined in XML and is categorized per system. The scope of the proposed mapping is restricted to the flight control, hydraulic, and electrical systems, respectively. However, the approach described can be adapted for other systems as well. The data schema consists of parent XML tags for each subsystem, where each tag consists of two top-level child tags, which are called “Inputs” and “Outputs”. The “Inputs” tag holds the system architecture inputs required by the system sizing tool as child tags, while the "Outputs" tag holds the parameters that are calculated by the system sizing tool.

The generic element descriptor shown in figure 3.13 represents the power view of a notional flight control actuation system architecture. It consists of two sources representing the engines of the aircraft and an emergency source representing a Ram Air Turbine (RAT). Furthermore, the architecture has three hydraulic distribution systems supplying three EHSAs associated with a single rudder surface. Shown on the right of the same figure is the section of the tags for the input parameters under the FCS tag. First, the "Control_Surface_func" tag is populated, iterating through all the nodes of the graph with "type: Device," reading the "function" property, and writing a "Control_Surface_func" tag for each control surface. Similarly, the graph can be queried to identify the nodes of "type:

Consumer" connected to each device element, and the "subtype" property of each consumer can be inspected to determine the actuator technology, which is then used to create a "Control_Surface" tag per device element. The actuator technology is listed in the tag according to the code corresponding to the actuator type (in this case, H1 is an EHSA); if there are multiple actuators allocated to a surface, they can be listed while separated by a semicolon.

Next, the graph is parsed for all nodes of "type: Distribution" and "subtype: Hydraulic"; a tag is created for each hydraulic distribution node and is named sequentially or according to the index of the corresponding graph node. The nodes are further queried to elicit inputs associated with the hydraulic distribution, such as hydraulic fluid, tubing efficiency, and associated pumps. Additionally, the hydraulic distribution nodes are parsed to determine which nodes are supplied by source nodes of "subtype: Engine" to determine with which engine a particular hydraulic distribution is associated.

Once the hydraulic system tags have been created, the association between the hydraulic system and the actuators is determined by checking the distribution nodes of "subtype: Hydraulic" connected to each actuator, represented by the consumer node. A similar approach is used for electrical actuators, where the "subtype" property being searched for is "Electrical". The identification of the hydraulic system determined in the previous step is applied to the "Control_Surface" tag, which itself is a child of the "Electrical_Hydraulic_Association" tag. In this manner, the graph can be parsed for information with which system sizing inputs can be provided to the MDAO workflow using a standardized data schema with a rich description of the system architecture.

3.4.2 Application: Integration of Safety Assessment into MDAO

Collaborative workflow

This section presents the integration of rule-based safety assessment into a collaborative MDAO workflow. The scope of the study presented here is twofold. First, this study aims to characterize the impact of safety rules derived from certification regulations at the system level. Second, it outlines an approach to integrate safety as a discipline within a collaborative MDAO workflow. The collaborative workflow in question is developed within the AGILE 4.0 project and consists of multiple disciplinary tools hosted by the collaborating organizations and integrated using the RCE platform as shown in figure 3.14. The system sizing tool used in this workflow is the ASTRID tool developed by the Politecnico di Torino. The workflow is used to study a 19-seater aircraft concept, which can either be certified to Part 23 or Part 25 regulations, and the certification constraints are evaluated using a certification tool.

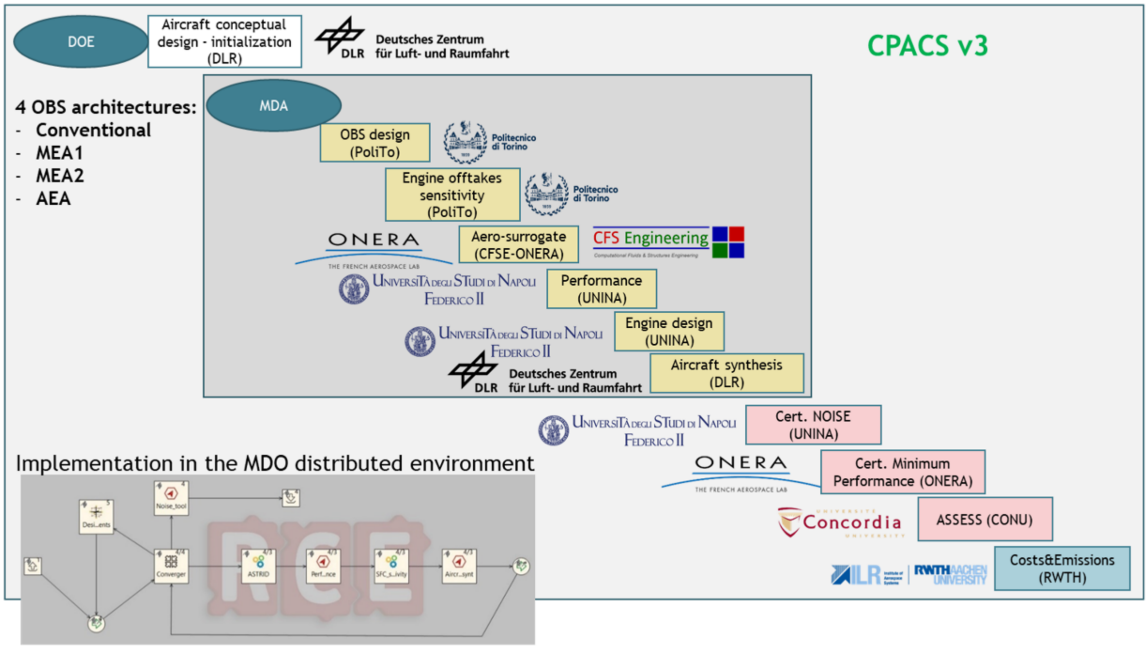

DOE
Aircraft conceptual design - initialization (DLR)
DLR Deutsches Zentrum für Luft- und Raumfahrt
CPACS v3
4 OBS architectures:
- Conventional
- MEA1
- MEA2
- AEA
MDA
OBS design (PoliTo)
Politecnico di Torino
Engine offtakes sensitivity (PoliTo)
Politecnico di Torino
ONERA THE FRENCH AEROSPACE LAB
Aero-surrogate (CFSE-ONERA)
CFS Engineering
Università degli Studi di Napoli Federico II
Performance (UNINA)
Università degli Studi di Napoli Federico II
Engine design (UNINA)
DLR Deutsches Zentrum für Luft- und Raumfahrt
Aircraft synthesis (DLR)
Università degli Studi di Napoli Federico II
Cert. NOISE (UNINA)
ONERA THE FRENCH AEROSPACE LAB
Cert. Minimum Performance (ONERA)
Concordia University
ASSESS (CONU)
ILR
RWTH AACHEN UNIVERSITY
Costs&Emissions (RWTH)
Implementation in the MDO distributed environment
RCE
Desi...ents
Noise_tool
Converger
ASTRID
Perf...nce
SFC_s...ivity
Aircr...synt

Figure 3.14: Collaborative workflow developed in the AGILE 4.0 project, from [75]

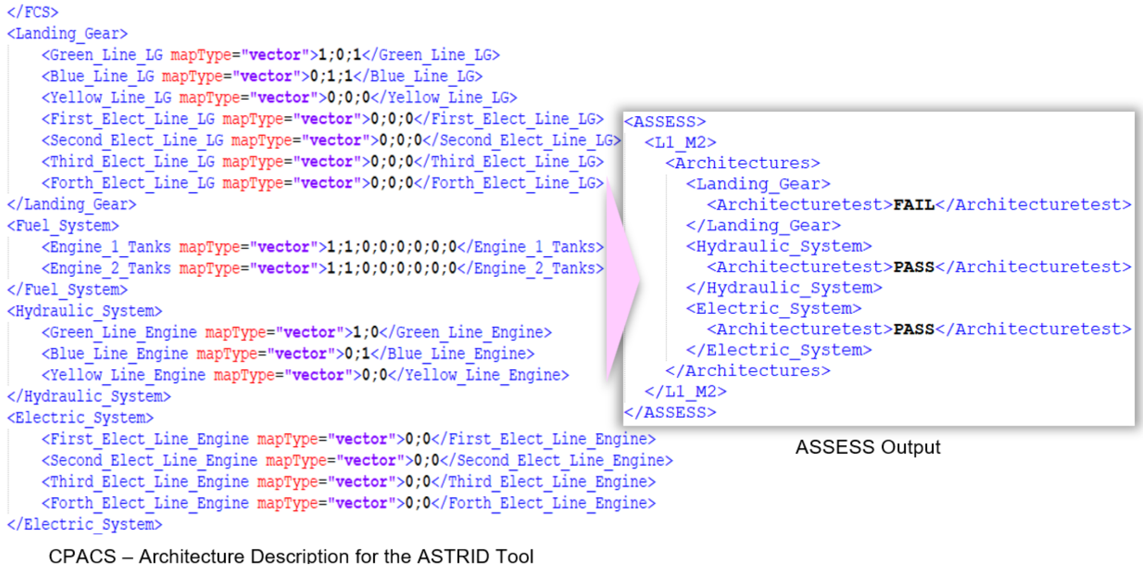
The ASSESS L0 module is adapted to work with the architecture description provided by the ASTRID system sizing tool. This study considered three architecture variants shown in table 3.3 that feature increasing levels of electrification across the flight control, environmental control, hydraulic system, and landing gear system.

Table 3.3: Architecture variants featuring increasing levels of system electrification

Architecture ID	Hydraulic Consumers	Electrical Consumers	Bleed-air Consumers
Baseline	FCS, LDG, Braking System	Avionics, Fuel, Galley, Lights, In-Flight Entertainment, etc.	ECS, IPS, No APU Bleed
MEA1	-	Avionics, Fuel, Galley, Lights, FCS, LDG, Braking System	ECS, IPS, No APU Bleed
MEA2	-	Avionics, Fuel, Galley, Lights, Local Electrically Powered Hydraulics (FCS, LDG, Braking System), ECS, IPS	-

The baseline architecture features hydraulic landing gear braking and hydraulically powered flight controls. The more-electric aircraft 1 (MEA1) architecture features the electrification of FCS and landing gear (LDG) braking functions, and the more-electric aircraft 2 (MEA2) architecture features local hydraulic power generation for FCS and LDG using electrical power. The bleed system is completely electrified such that the Environmental Control System (ECS) and Ice Protection System (IPS) are electrically powered. A complete description of each architecture is provided in [75, p. 4].

The landing gear braking system is specified using a standardized data schema with a mixed fidelity description of the system architecture. Here, the ASTRID tool's input-output data model is integrated into the tool-specific section of CPACS. A snippet of the system architecture descriptor is shown in figure 3.15, where the "Landing_Gear" tag outlines the different power supplies allocated to the landing gear using a unique tag for each hydraulic power system. Each hydraulic system contains a vector of three elements, with each element being assigned a value of 0 or 1. The vector represents the nose and the two main landing gear braking units, respectively. For example, the "Green_Line_LG" tag specifies a vector of "1;0;1". This implies that the "Green" hydraulic system supplies the nose and right main landing gear braking units. Similarly, the "Green_Line_Engine" tag specifies which engine provides the secondary power to pressurize the green hydraulic system.



```
</FCS>
<Landing_Gear>
    <Green_Line_LG mapType="vector">1;0;1</Green_Line_LG>
    <Blue_Line_LG mapType="vector">0;1;1</Blue_Line_LG>
    <Yellow_Line_LG mapType="vector">0;0;0</Yellow_Line_LG>
    <First_Elect_Line_LG mapType="vector">0;0;0</First_Elect_Line_LG>
    <Second_Elect_Line_LG mapType="vector">0;0;0</Second_Elect_Line_LG>
    <Third_Elect_Line_LG mapType="vector">0;0;0</Third_Elect_Line_LG>
    <Forth_Elect_Line_LG mapType="vector">0;0;0</Forth_Elect_Line_LG>
</Landing_Gear>
<Fuel_System>
    <Engine_1_Tanks mapType="vector">1;1;0;0;0;0;0;0</Engine_1_Tanks>
    <Engine_2_Tanks mapType="vector">1;1;0;0;0;0;0;0</Engine_2_Tanks>
</Fuel_System>
<Hydraulic_System>
    <Green_Line_Engine mapType="vector">1;0</Green_Line_Engine>
    <Blue_Line_Engine mapType="vector">0;1</Blue_Line_Engine>
    <Yellow_Line_Engine mapType="vector">0;0</Yellow_Line_Engine>
</Hydraulic_System>
<Electric_System>
    <First_Elect_Line_Engine mapType="vector">0;0</First_Elect_Line_Engine>
    <Second_Elect_Line_Engine mapType="vector">0;0</Second_Elect_Line_Engine>
    <Third_Elect_Line_Engine mapType="vector">0;0</Third_Elect_Line_Engine>
    <Forth_Elect_Line_Engine mapType="vector">0;0</Forth_Elect_Line_Engine>
</Electric_System>
```

CPACS – Architecture Description for the ASTRID Tool

```
<ASSESS>
  <L1_M2>
    <Architectures>
      <Landing_Gear>
        <Architecturetest>FAIL</Architecturetest>
      </Landing_Gear>
      <Hydraulic_System>
        <Architecturetest>PASS</Architecturetest>
      </Hydraulic_System>
      <Electric_System>
        <Architecturetest>PASS</Architecturetest>
      </Electric_System>
    </Architectures>
  </L1_M2>
</ASSESS>
```

ASSESS Output

Figure 3.15: Architecture description in ASTRID and corresponding ASSESS output written to the "toolspecific" tag in CPACS

The aim of this study was to evaluate the impact of the Part 23 and Part 25 certification basis at the aircraft level. However, at the system level, the difference between the two categories of regulations manifests itself in the application of safety heuristics governing the minimum redundancies required in power generation and distribution systems and their allocation to the aircraft landing gear braking systems. Redundant generation and distribution systems then contribute to aircraft-level weight, which is used to assess the impact of the certification basis. This effect can have an impact on the feasibility of smaller Part 23 aircraft that are being considered for integration of hybrid-electric and distributed electric propulsion technologies. The additional weight penalties associated with the required redundancies for new technologies that are applied to critical aircraft functions can result in an aircraft that would need to be re-certified under the Part 25 category, thereby increasing overall cost and feasibility. Although this study does not directly deal with the implementation of novel technologies, it serves to demonstrate how redundancy requirements vary according to certification category.

Originally, one of the objectives of the study was to integrate safety by evaluating multiple system architectures using the developed heuristics. However, the architecture design space is constrained to only four system architectures; the integration of ASSESS

L0 was modified to evaluate all four architecture variants and evaluate the weight of the hydraulic, electrical, and landing gear braking system for each architecture and provide a weight differential compared to the weight of the hydraulic, electrical, and landing gear braking systems of the baseline aircraft. The weights were determined by passing the system architecture descriptor to the ASSET tool, followed by the computation of the weight deltas, which were then written as the output of the ASSESS L0 SysArc tool along with the typical "Pass" or "Fail" output provided by the tool. Table 3.4.2 shows the overall weight deltas obtained for each system architecture and each certification basis.

Table 3.4: Comparison of hydraulic and electrical systems across different certification bases

System	**No. of Assigned Hydraulic Lines**	**No. of Assigned Electrical Lines**	**Certification Basis**	**Hydraulic System Delta (kg)**	**Electrical System Delta (kg)**
Hydraulic Landing Gear Braking	1	1	Part 23	0	0
	2	2	Part 25	34.1	0
MEA 1	0	2	Part 23	-34.6	25.1
	0	3	Part 25	-34.6	284.8
MEA 2	0	2	Part 23	-34.6	53
	0	3	Part 25	-34.6	306.3

The results of this study indicate that both the MEA 1 and MEA 2 options benefit from the weight reduction obtained by eliminating the hydraulic system. MEA 1 and MEA 2 architectures both exhibit an increase in electrical system weight stemming from the greater number of redundancies required for Part 25 certification compared to Part 23 certification. However, among both MEA 1 and MEA 2, the former incurs a lower electrical system weight increment of the two. Thus, MEA 1 appears promising for system-level weight reduction potential, subject to an aircraft-level characterization of the system weight reduction on MTOW, fuel burn, and other metrics. Overall, this study has shown that safety rules can be applied in a collaborative workflow even using a mixed fidelity descriptor; it outlines a strategy to evaluate the impact of safety rules derived from certification regulations on system sizing, thereby enabling a link between safety, certification, and aircraft-level weight evaluation.

Industrial workflow

The integration of safety analysis with the industrial MDAO workflow, using the methods described in section 3.4.1, is the basis of the following study. Here, as shown in figure 3.16, a hydraulically powered yaw control actuation architecture is specified using the spreadsheet-based descriptor. It is important to note that the number of actuators, the number of surfaces, and the number of hydraulic distributions are fixed using the safety rules. The APU flag in the spreadsheet-based descriptor is activated, and the actuation type is set to EHSA. The only possible variation permitted in the baseline architecture is the presence or absence of a power transfer unit.

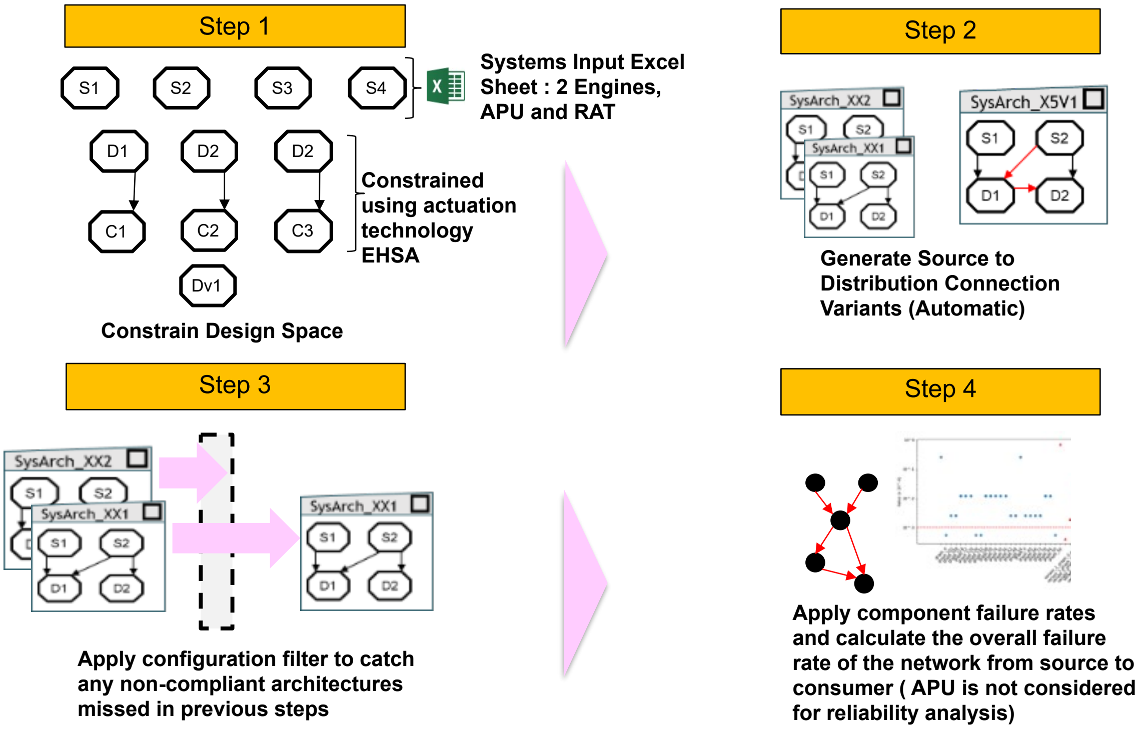

Step 1
S1
S2
S3
S4
Systems Input Excel Sheet : 2 Engines, APU and RAT
D1
D2
D2
C1
C2
C3
Dv1
Constrained using actuation technology EHSA
Constrain Design Space
Step 2
SysArch_XX2
SysArch_XX1
SysArch_X5V1
S1
S2
D1
D2
Generate Source to Distribution Connection Variants (Automatic)
Step 3
SysArch_XX2
SysArch_XX1
S1
S2
D1
D2
Apply configuration filter to catch any non-compliant architectures missed in previous steps
Step 4
Apply component failure rates and calculate the overall failure rate of the network from source to consumer (APU is not considered for reliability analysis)

Figure 3.16: Generation of a constrained design space

This baseline architecture, as specified in the spreadsheet descriptor, is used to automatically generate the relevant generic elements. The distribution elements are connected to the consumer elements based on the actuation technology employed, i.e., a consumer element only receives a connection from one distribution element at a time. A combinatorial design space of candidate architectures is defined based on specifying connections between the source elements and distribution elements as well as connections between distribution elements themselves. The latter are termed variants as they differ from existing architectures in the list by a connection between distribution elements. Each architecture denoted by "A" followed by a number can have a variant denoted by "A" followed by the architecture number, and then V followed by the variant number. Therefore, an architecture A1 can have variants A1V1, A2V2, etc., where each variant will have a different connection between the distribution elements but will resemble the parent architecture in all other connections between constituent elements.

Once the candidate architectures are generated, they are first filtered using the safety and configurational rules to remove ill-configured architectures. The remaining architectures are then evaluated using the simple path approach to determine the overall system failure rate - a measure of the redundancy in the architecture - for an exposure time of one hour. The architectures that meet the safety requirement of 10^{-9} failures per flight hour (FH) are retained for further analysis. As figure 3.17 shows, these are variant architectures featuring connections between distribution systems, which, when written to the spreadsheet descriptor, result in the PTU flag being set to active. Further evaluation of each of the safety characteristics of each architecture can then be performed using the quantitative safety assessment methods using the generic element descriptor that is defined in section 3.6.2.

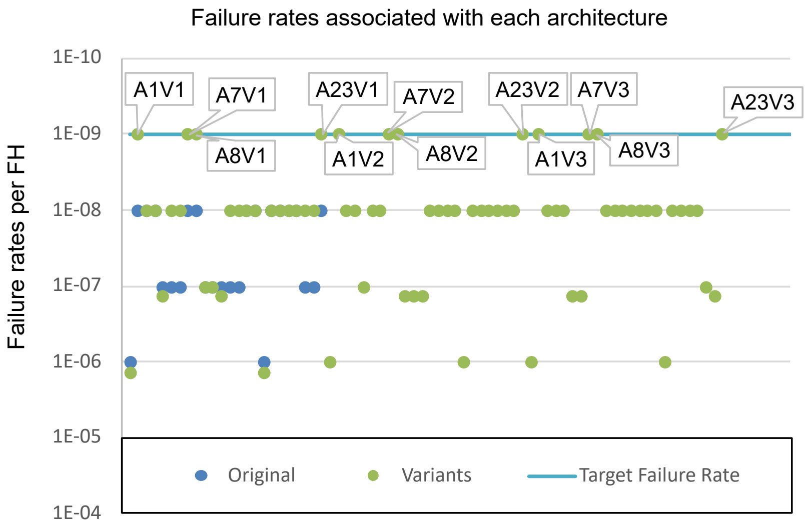


Figure 3.17: Failure rates evaluated for each architecture in the design space

This example demonstrates several important aspects of the generic element descriptor that enables the integration of safety into an existing industrial MDAO workflow. First, the simplified representation of the generic descriptor provides sufficient information from which to derive system architecture inputs that are required by system sizing methods integrated into an MDAO workflow. In the provided example, a single rudder surface is used, and a hydraulic actuation system is considered. However, the graph descriptor can also be used to represent unconventional architectures (see section 4.3). The generic element descriptor can therefore enable the evaluation of unconventional architectures using the industrial workflow, provided suitable modifications are made to the system sizing methods themselves.

Second, the automated generation of the relevant generic elements from a baseline architecture and the synthesis of a system architecture design space is enabled by using the generic element descriptor. The graph-based format of the descriptor enables it to be used for the evaluation of safety rules, even if the number of architectures or the design space is large. Furthermore, even quantitative safety evaluation can be performed at this stage to filter out unfeasible architectures and provide insight into the safety characteristics of the remaining candidate architectures. This results in system architecture input provided to the system sizing tools that specify the required system redundancies in a safety-informed manner. The resulting impact of the system architecture weight and power draw at the aircraft level (the former of which is directly impacted by redundancy requirements) can benefit from improved confidence in system architecture feasibility and a potential reduction in epistemic uncertainty. The former affects the likelihood of additional weight penalties being incurred later in the design process, while the latter can affect the confidence in the evaluated aircraft performance.

Finally, even if the architect does not wish to perform any architecture trade studies or MDAO-driven aircraft-level optimization while considering multiple architecture options that have passed both qualitative and quantitative safety checks, the system architect, by defining a single architecture using the generic element representation, can already evaluate its safety characteristics, thereby aiding decision making in the system architecting process. If the architect then wishes to select an architecture and provide it to the system sizing tool of the industrial MDAO workflow, this can easily be done using the link between the graph descriptor and the input descriptor of the industrial workflow.

3.5 Integrating ASSESS-L0 with an MBSE environment

ASSESS L0 SysArc deals with the initial exploration of the system architecture design space and ensures that candidate architectures that pass the safety rule-based filtering meet basic redundancy requirements. This enables the system architect to analyze these architectures in an MDAO-based evaluation environment and develop them further with additional confidence in their feasibility from a safety and performance perspective. A formal system architecture specification is required to perform further safety assessments, such as the model-driven FHA, and to capture system interfaces for development in the preliminary design stage. Furthermore, developing a system architecture specification model in conceptual design ensures that the architect and the safety analyst have a common understanding of the system early in the design process.

The development of a specification model using the graph descriptor consists of three different steps which are:

1. Element mapping
2. Data serialization and transfer
3. Specification model instantiation

The mapping between the graph descriptor and the MBSE specification model is based on an element-to-element correspondence. A key assumption in building an initial specification model from the graph descriptor is that a catalogue of system functions and logical components based on the approach described in [32] is implemented in the MBSE environment. The mapping then instantiates a corresponding component in the MBSE environment for each component in the graph descriptor. The instantiated components already have generic functions allocated to them based on the pre-existing catalogue of elements.

In this work, the Capella MBSE environment is used extensively, primarily due to its focus on functional analysis and function-based system architecture development. This philosophy is compatible with the overall functional analysis-driven system development and safety assessment outlined in the ARP4761 and ARP4754A. Furthermore, the methods outlined in this thesis were developed within the context of a research project with an industry partner and, as such, the use of Capella is driven by the experience and preferences of the industry partner.

Element mapping

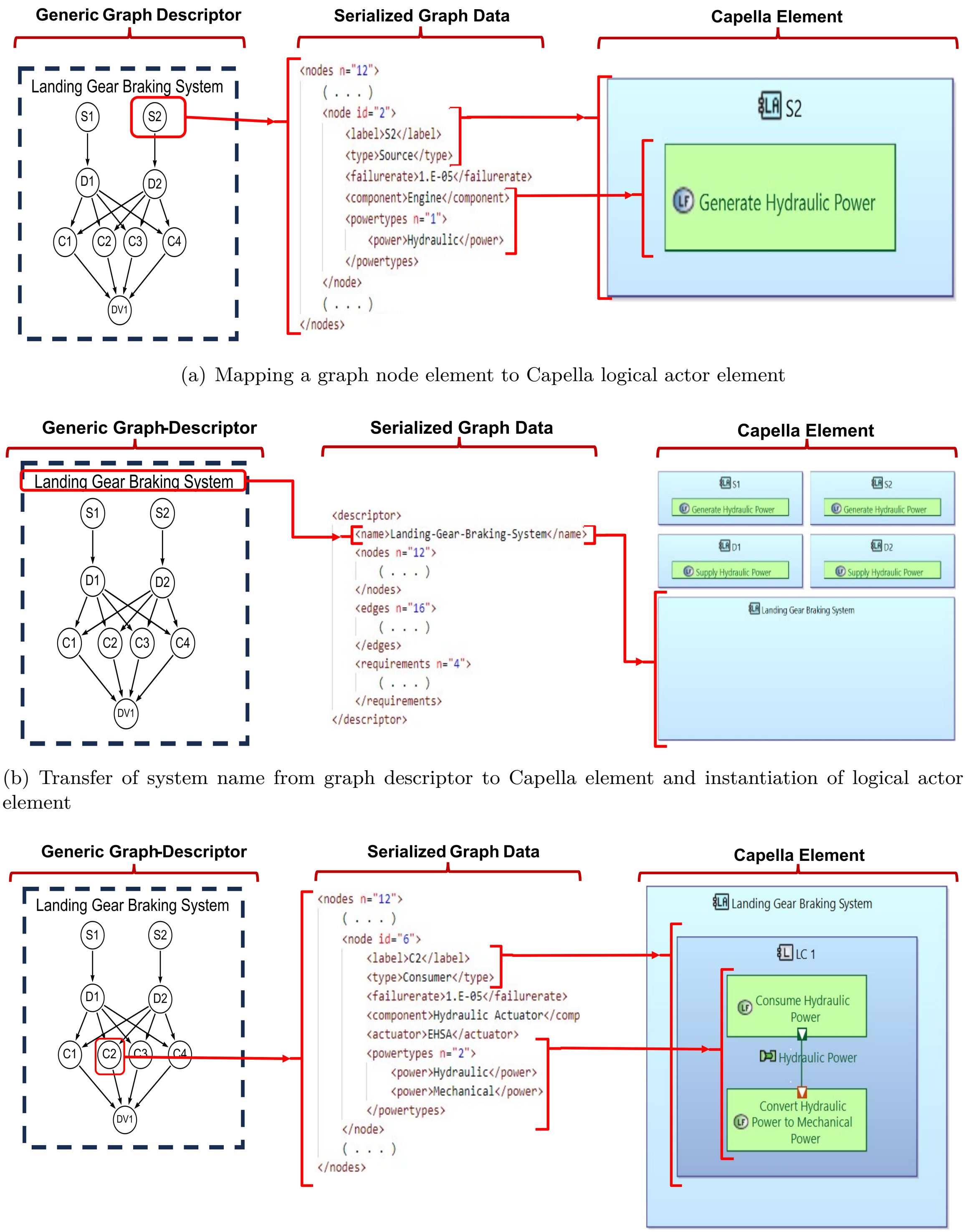
The elements of the graph descriptor are mapped to logical components in the Capella model. The elements of the graph descriptor are mapped to the Capella model and represented at a specific level. The mapping employs a direct element-to-element correspondence, i.e., each element in the graph descriptor - represented by the nodes of the graph - is instantiated as a logical component in Capella. The properties of the generic element embedded in the graph node are also transferred to Capella and stored as properties of the logical component using the Property Values Management Toolkit (PVMT) plugin in Capella. The generic elements in the descriptor can be of various types, i.e., Source, Distribution, Consumer and Device – thus, in the Capella model, these are referenced in the name of each corresponding element. Figure 3.18(a) shows how each node corresponds to a logical element in Capella, and figure 3.18(b) illustrates how the logical components are named.

The power and control flows exchanged between elements are also mapped directly from the descriptor as component exchanges between logical components in Capella. However, the catalogue of generic logical elements in Capella already have generic functions assigned to them, and these functions have predefined functional exchanges. Figure 3.18(c) shows how a logical component from the catalogue already has functions assigned to it along with existing functional exchanges that are carried over from the element catalogue. The element catalogue is defined using the REC/RPL feature in Capella.

Data serialization and transfer

Data serialization refers to the parsing of the generic element descriptor and the collection of relevant data to be transferred to the MBSE environment. The generic element descriptor is implemented as a network graph, and the serialization process begins by parsing through each node of the graph and reading the node's properties. Each node represents a generic element, and the properties are structured and read hierarchically, starting with the name or label of the node. This is followed by extracting the "type" property, which indicates whether the node is a Source, Distribution, Consumer or Device. These are the primary properties stored in each graph node. Other properties can include sub-component allocations represented by "subtype" or component-specific data such as failure rate or weight.

In addition to reading node properties, the serialization process extracts information on the connections between each generic element. These connections are represented as edges between the nodes of a network graph. As each node is parsed, the number of incoming and outgoing edges associated with each node is identified. This is done on the basis of the label and the identification parameter of the node. The edges also store a "powertype" and "controltype" property, which indicates the nature of the power or control flow.



```
<nodes n="12">
  ( . . . )
  <node id="2">
    <label>S2</label>
    <type>Source</type>
    <failurerate>1.E-05</failurerate>
    <component>Engine</component>
    <powertypes n="1">
      <power>Hydraulic</power>
    </powertypes>
  </node>
  ( . . . )
</nodes>
```

(a) Mapping a graph node element to Capella logical actor element

```
<descriptor>
  <name>Landing-Gear-Braking-System</name>
  <nodes n="12">
    ( . . . )
  </nodes>
  <edges n="16">
    ( . . . )
  </edges>
  <requirements n="4">
    ( . . . )
  </requirements>
</descriptor>
```

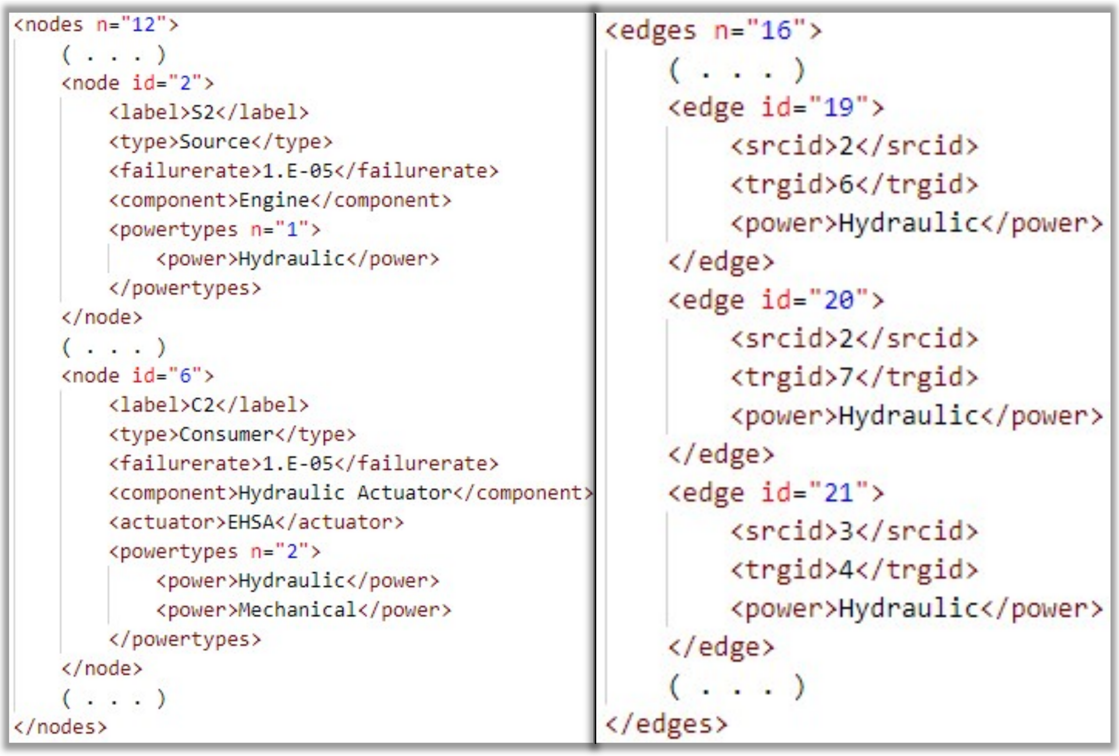
(b) Transfer of system name from graph descriptor to Capella element and instantiation of logical actor element

```
<nodes n="12">
  ( . . . )
  <node id="6">
    <label>C2</label>
    <type>Consumer</type>
    <failurerate>1.E-05</failurerate>
    <component>Hydraulic Actuator</comp
    <actuator>EHSA</actuator>
    <powertypes n="2">
      <power>Hydraulic</power>
      <power>Mechanical</power>
    </powertypes>
  </node>
  ( . . . )
</nodes>
```

(c) Mapping a consumer element to a corresponding logical component in Capella

Figure 3.18: Mapping between graph descriptor and Capella model elements

The information extracted by parsing each node and edge of the graph descriptor is stored in a data serialization file. This is an XML file structured to capture the information extracted from the graph. It uses the node identification parameter and node labels to store information in specific tags for each node. Each tag has a special identification string that is used to reference the edge information, which is stored in a separate tag.



```
<nodes n="12">
    ( . . . )
    <node id="2">
        <label>S2</label>
        <type>Source</type>
        <failurerate>1.E-05</failurerate>
        <component>Engine</component>
        <powertypes n="1">
            <power>Hydraulic</power>
        </powertypes>
    </node>
    ( . . . )
    <node id="6">
        <label>C2</label>
        <type>Consumer</type>
        <failurerate>1.E-05</failurerate>
        <component>Hydraulic Actuator</component>
        <actuator>EHSA</actuator>
        <powertypes n="2">
            <power>Hydraulic</power>
            <power>Mechanical</power>
        </powertypes>
    </node>
    ( . . . )
</nodes>
```

```
<edges n="16">
    ( . . . )
    <edge id="19">
        <srcid>2</srcid>
        <trgid>6</trgid>
        <power>Hydraulic</power>
    </edge>
    <edge id="20">
        <srcid>2</srcid>
        <trgid>7</trgid>
        <power>Hydraulic</power>
    </edge>
    <edge id="21">
        <srcid>3</srcid>
        <trgid>4</trgid>
        <power>Hydraulic</power>
    </edge>
    ( . . . )
</edges>
```

Figure 3.19: Serialization of graph node and edge data from a generic element descriptor

Figure 3.19 illustrates an example of how the information parsed from the graph nodes have their own tags in the XML file and how the edge information and edge properties are stored in a separate tag but are referenced to their parent graph nodes using a unique identifier.

Specification model instantiation

The information extracted from the graph is used to instantiate Capella model elements and build a specification model. From a methodological perspective, the relevant information required to create a system architecture specification is the number and type of each component and the power and control exchanges between each component. This information can be determined by reading the type properties and identifying the connections between components and the nature of those connections. Using the data serialization process, all the required information is stored in the serialization XML file.

Creating a system architecture specification model in an MBSE environment from the generic element descriptor relies on the generic element catalogue of functions and logical elements that need to be created. Furthermore, the generic logical elements need to be modelled at the appropriate level of granularity to ensure that the specification model remains at a manageable level of detail for conceptual design. The generic elements catalogue proposed in Jeyaraj [32] is further extended into a formal MBSE modelling framework by Tabesh [68] and prescribes multiple levels of granularity for systems specification modelling in conceptual design.

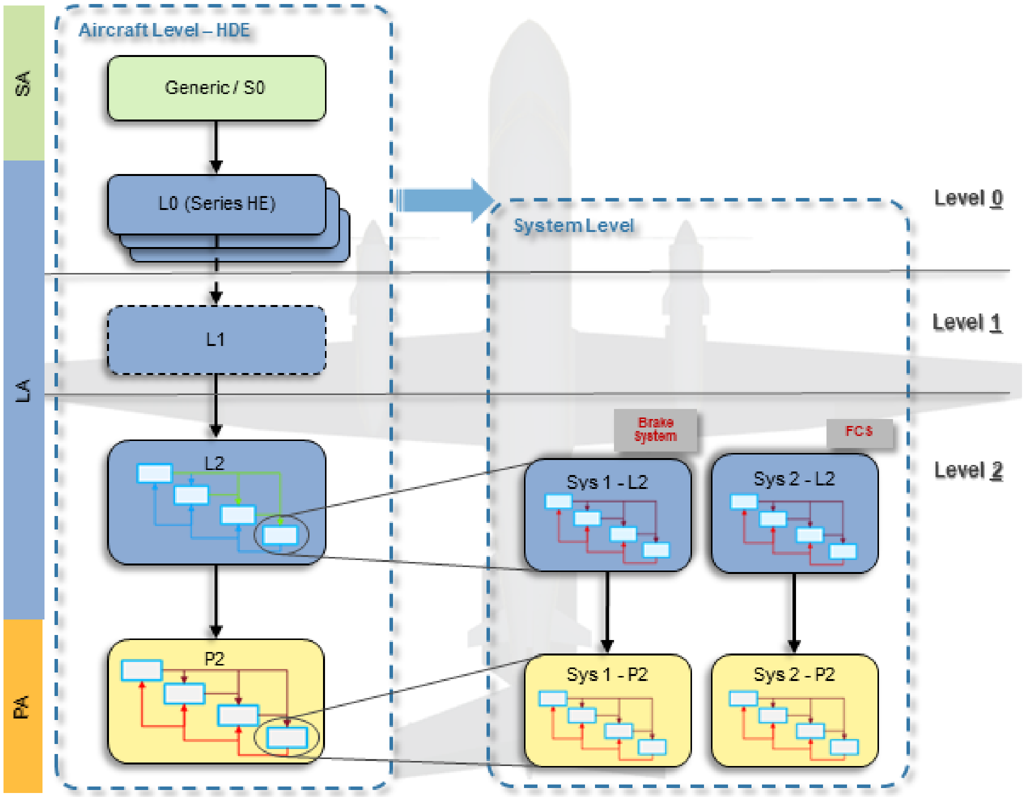


Figure 3.20: MBSE specification framework, from [68]

The framework shown in figure 3.20 comprises multiple levels of modelling detail and distinguishes between the level of detail required at the aircraft level and the system level. The framework prescribes a generic aircraft-level model of the system architecture called S0, and the generic functions discussed earlier are modelled at the S0 level. The generic S0 functions are then allocated to logical components at the L1 level. The L1 level is earmarked specifically to represent the logical architecture with component, power-path, and control-path redundancies.

3.6 ASSESS L1-M1

The scope of ASSESS L1-M1 lies in further analysis of a limited set of candidate system architectures that have passed earlier safety checks and evaluation using system sizing tools. These architectures are promising from a performance and weight perspective but could benefit from a deeper understanding of their safety characteristics. ASSESS L1-M1 seeks to specify safety targets and also evaluate the system architecture using both qualitative and quantitative techniques. The following analyses are a part of ASSESS L1-M1:

1. Functional Hazard Assessment
2. Quantitative Safety Assessment

The functional hazard analysis serves as the entry point to the formal safety assessment process defined by ARP 4761. The FHA serves to identify functional failures, categorizes them based on the severity of impact on the aircraft, and sets safety targets for each function. The systems implementing those functions then inherit safety targets related to those failures. The FHA is a top-down approach and lies on the specification arm of the aircraft development process.

FHAs are typically paper based, although there has been increasing interest in performing them using MBSE tools to benefit from formal representations of the system and to capture safety information in the system model. These are termed Model-based FHAs (MB-FHA) and focus on developing methods to capture information from the output of an FHA in the model. This is usually done by developing custom profiles in an MBSE tool with artifacts that can be associated with model elements; these elements contain fields that can be filled in with information generated while performing the FHA. These approaches consider the system modelling and FHA processes to be disparate. This dichotomy leads to an information gap between two disciplines at play – systems architecting and safety assessment, thereby leading to multiple iterations between the architect and the safety analyst.

In this context, the architect typically specifies the system model, and the safety analyst compiles the system functions, examines interactions, and analyses failure conditions. Failure conditions are often analyzed by building functional hierarchies and determining the interaction between functions across the aircraft and system levels. This is an important and often complex part of the FHA process, as functional failures and failure combinations can impact multiple functions at the aircraft level. The system architect views chains of functional interaction as a series of success relationships, whereas the safety analyst can have a different perspective and view them as failure relationships.

This thesis posits that existing MB-FHAs are useful approaches in cases where the system architecture is established, the functions and their interactions are known, and similarity with past experience and knowledge bases can be relied upon. However, for conceptual design, MB-FHAs are not flexible enough. The conceptual design stage can accommodate the exploration of different options and the specification of novel functional architecture; as a result, in the context of the FHA, a means of providing better safety insight to the system architect is desirable. If the system architect already has the tools to be able to analyze the architecture they are building for safety, they are then better positioned to incorporate safety considerations into the architecture as it is being developed.

At the same time, if the system architect can represent their safety analysis in a manner familiar to the safety analyst, they can then benefit from the safety analyst's expertise early in the design process. Iterations will exist between the system architect and the safety analyst, but the process of evaluating safety as the architecture is being built and having the analysis reviewed in an effective manner has the potential to improve the overall understanding of the system architecture. These activities, when performed in an MBSE environment, capture the output of the FHA in the system model, which is made available for further analysis in subsequent design stages. This model-driven FHA provides improved insight into the safety characteristics of the system architecture and can enable the system architect to modify the architecture in response to emerging safety characteristics.

3.6.1 Model-driven Functional Hazard Assessment (MdFHA)

The MdFHA process consists of simultaneously modelling and analyzing the functional and logical architecture specifications from a safety perspective in an MBSE environment and storing the resulting safety information within the specification model. An MBSE environment consists of a modelling language and a tool. The MdFHA approach is developed to be tool agnostic, but this thesis describes the approach using the ARCADIA framework implemented using the Capella MBSE tool. The choice of ARCADIA/Capella was driven by three factors. The first is the preference and experience of the industry partner in this research, Bombardier, in using Capella. The second is to build upon the framework that was developed in prior work. The third factor is that the ARCADIA methodology's basis in the functional analysis [149] and function-based system architecture specification lends itself well to application in the function-based development and safety assessment processes defined in the SAE ARP4754A and SAE ARP4761, respectively.

Contribution statement for the MdFHA method

The MdFHA method described in this thesis is developed through a collaborative effort and is documented in [150]. The contributions of this thesis to the development of the method are as follows:

1. Conceptualizing the approach for modelling the system while simultaneously analyzing the system under development to inform the development of the FHA
2. Identifying Capella diagrams to be used to analyze the system architecture to generate inputs to the FHA
3. Developing a storage and visualization strategy to store the outputs of the FHA within system model artifacts

The MdFHA approach consists of three key elements as listed below:

1. System modelling and establishing the appropriate level of granularity
2. Functional Hazard Analysis using Capella diagrams
3. Storing FHA results in the system model

System modelling and granularity

The system architecture can be modelled at multiple levels of granularity and functional analysis, as specified by the ARP4754A, which results in functional decomposition and the linking of functions across the aircraft and system levels. Aircraft-level functions are abstract and focus on the high-level description of what the aircraft needs to achieve. System-level functions are more granular and feature additional detail; often, several system functions can be developed from the decomposition of a single aircraft-level function, depending on the desired level of detail. The interactions between system functions are also important features that need to be captured, as these can have a logical relationship with the parent aircraft-level function. Typically, the FHA starts with an aircraft-level description of functions, which are then allocated to systems. System-level functions are developed by the decomposition of top-level aircraft functions.

A system architect could start modelling the functional architecture at any level. However, an unstructured approach to system modelling may cause ambiguous representations of the architecture and inhibit the ability of the model to be used as a central source of information for downstream activities. Alternatively, the system architect could expend significant effort in developing a detailed model of the system architecture, but this could extend beyond the scope of systems architecting within the conceptual design phase and has the risk of locking in a specific architecture or technology choice, thereby precluding the possibility of architecture exploration. Thus, guidance in the form of a modelling framework can enable the architect to purposefully develop the system architecture specification at different modelling levels based on the end application of the model.

The modelling language and tool choice can also emphasize the need for structured guidance material for system specification development. The ARCADIA framework specifies four levels at which the system can be modelled: the operational, system, logical, and physical levels. Each level has a specific purpose. The operational level features an operational analysis, which helps the architect model the expected operational context of the model and helps identify the key activities that the system needs to perform. The system level features the system analysis, which enables the architect to specify system functions that will be involved in realizing the operational context of the system of interest. The logical level deals with the specification of logical components and the allocation of system functions to these components. From the perspective of the FHA, the operational analysis can be neglected, especially if some decisions on system technology or system scope have already been made. However, considering just the system and logical levels, the system architect is free to model at any level of detail. Finally, implementing MBSE specification modelling in an industrial context is challenging as it typically contends with incumbent approaches and ways of decomposing system architectures. This further emphasizes the value of a structured modelling framework that specifies different levels of granularity and is adapted to the modelling framework of choice – in this case, ARCADIA.

The entry point into the definition of the system architecture specification model starts with the S0 description (see figure 3.20) of the functional architecture using generic functions. These generic functions are allocated to logical components to build the L0 logical specification of the system architecture. The L0 logical specification enables the system architect to characterize the interactions between system functions and, as a result, define connections between logical components. Each logical component typically represents a subsystem, enabling the system architect to decompose the generic aircraft-level functions into system-level functions.

From the L0 logical specification, the architect can proceed directly to the L2 logical level if the system architecture being developed is conventional. The L1 logical level is a transitional modelling level that is used to capture the interface between aircraft and system-level functions, specifically for unconventional system architectures. At the L1 level, an unconventional system architecture is intended to be gradually developed through interactions between a system architect and a safety expert. An example of an unconventional system architecture, namely a hybrid-electric propulsion system and its interaction with other aircraft systems, captured using the proposed modelling framework, is detailed in [68].

The L1 model contains all the key system components and system-level functional interactions. The system architect has the flexibility to go from L0 to a Sys-L1 model or to an L2 model. In a Sys-L1 model, the system architect can develop an individual subsystem in greater detail. At the L2 logical level, the system architect can add component redundancies and allocate the system-level functions to these logical components. The ARCADIA approach ensures traceability between all the prescribed modelling levels from S0 to P2. The detail subsystem models developed at Sys-L1, Sys-L2, and Sys-P2 are also traceable. However, it should be noted that once the physical layer is initiated, diagrams 5 and 6 are used to build the functional and logical architectures; these include the collation of system functions and interactions as well as the allocation of logical functions to logical components. The diagrams within Capella that are used in the proposed modelling framework are as follows:

1. System Level: [SFBD] - System Functional Breakdown
2. Logical Level: [LFBD] - Logical Functional Breakdown
3. System Level: [SDFB] - System Dataflow Blank
4. Logical Level: [LDFB] - Logical Dataflow Blank
5. System Level: [SAB] - System Architecture Blank
6. Logical Level: [LAB] - Logical Architecture Blank
7. Scenario: [FS] - Functional Scenario
8. Modes and States: [MSM] - Modes and States Diagram

Diagrams 1-4 are used to specify functions, transition functions from the system to the logical level, build system and logical function hierarchies, and identify functional interactions, respectively. Diagrams 5 and 6 are used to build functional and logical architectures, respectively, which include the collation of system functions and interactions as well as the allocation of logical functions to logical components. Diagrams 7 and 8 focus on developing specific scenarios of functional chains and categorizing different modes during which specific functions are active, respectively.

Functional Hazard Analysis using Capella diagrams

The Functional Hazard Assessment (FHA) serves as the formal entry point into the ARP4761 safety assessment process. It begins with the preliminary aircraft safety assessment, which involves the development of the Aircraft FHA (AFHA). The AFHA involves the identification of aircraft functions and the analysis of aircraft-level functional failures, leading to a categorization of functional failure severity as Catastrophic, Hazardous, Major, and Minor in decreasing order of severity. This analysis sets safety targets for the failure of top-level aircraft functions. The AFHA is followed by the derivation of system-level functions from the aircraft functions. These functions are organized and subject to the System FHA (SFHA), which further classifies the impact of system function failure conditions on the top-level aircraft functions. The AFHA and SFHA have the following aspects in common:

1. Specifying function hierarchies
2. Tracing the impact of functional failures
3. Classifying the severity of functional failures on aircraft-level functions

All the aforementioned activities involved a specification of functional hierarchy and an analysis of the relationship between functions, both at the same level and across aircraft and system levels. This is typically done using a paper-based process and relies on both visual and textual artifacts that represent the system architecture. The Capella workbench can be used to model the architecture specification and analyze the relationships between functions using a diverse set of diagrams. The MdFHA presented here will use specific diagrams in

Capella to support the architect in the process of analyzing the system architecture to conduct the FHA.

The MdFHA process is developed to account for the following three potential aircraft development scenarios:

1. The system functions, decomposition and hierarchies are available and common between aircraft programs

2. A baseline set of system functions is used, with additional functions being added for a specific aircraft program

3. A novel architecture is being developed where the majority of the functions are new or highly integrated

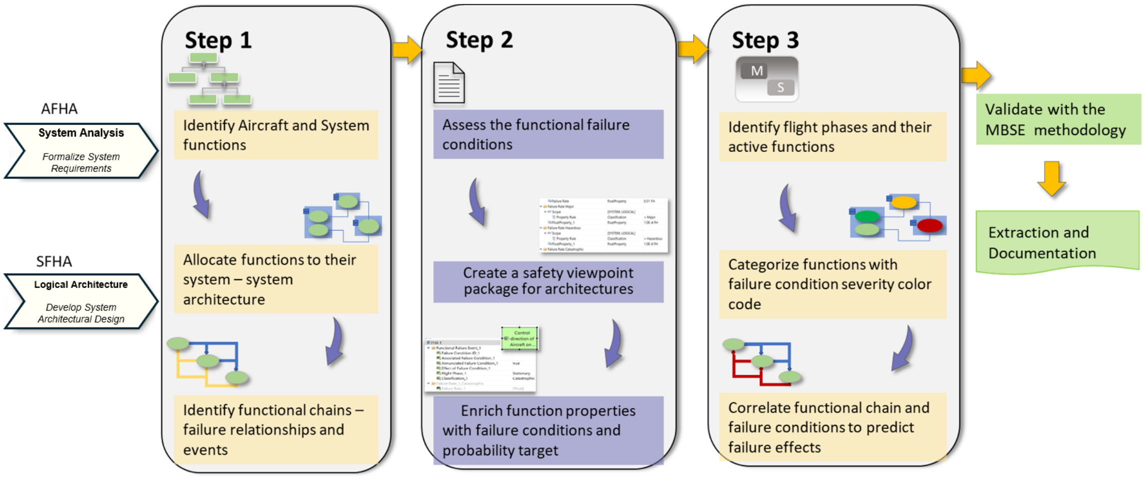

AFHA
System Analysis
Formalize System Requirements
SFHA
Logical Architecture
Develop System Architectural Design
Step 1
Identify Aircraft and System functions
Allocate functions to their system – system architecture
Identify functional chains – failure relationships and events
Step 2
Assess the functional failure conditions
Create a safety viewpoint package for architectures
Enrich function properties with failure conditions and probability target
Step 3
M
S
Identify flight phases and their active functions
Categorize functions with failure condition severity color code
Correlate functional chain and failure conditions to predict failure effects
Validate with the MBSE methodology
Extraction and Documentation

Figure 3.21: Overview of the formalized MdFHA integration within Capella from [68]

Figure 3.21 shows the steps involved in specifying and analyzing a system architecture in Capella, followed by storing the results of the FHA. The process is grouped into three buckets based on where the system architect starts. If the architect is starting from an existing baseline model of the architecture in Capella, then step 2 is the starting point. However, if the architect is working with functions that are different or with an entirely novel architecture, then the activities in step 1 must be followed. The following section will detail the activities and Capella diagrams involved within each step, with the example of the aircraft landing gear braking system. Here, the landing gear braking system architecture and function naming conventions of the ARP4761 braking system example are used [21, p. 174], although in the example shown in [150] the functional architecture will be built from the ground up.

Specification of function hierarchies

The first activity in step one is identifying the aircraft and system-level functions and modelling them in Capella. The functional breakdown diagram in Capella is used at the system analysis level to model the aircraft-level functions and assign the system-level functions to the functional hierarchy. The second activity is to build an SFDB in which the interactions between system functions are clearly defined. The system architecture blank can then be used to develop a unified view of the system architecture using system-level functions. Using the SAB or the SFDB, individual functional failures can be examined by identifying failure relationships that result from that functional failure. Failure relations can be modeled using the functional chain capability in Capella and can be visualized directly in the SAB, in the SFDB, or by using the dedicated functional chain diagram.

Tracing the impact of functional failures

The MdFHA approach uses the functional data flow, functional chain, and scenario diagrams to analyze functional failures and to outline failure relationships and effects. The functional chain diagrams used in this MdFHA process are repurposed and should be understood to represent a functional failure relationship, or rather, negative or unsuccessful relationships. Identifying these failure relationships and failure effects can help the system architect and safety expert interact to classify the severity of the failure of system functions, which is the main outcome of the FHA.

Storing FHA results in the model

An add-on for Capella called the Property Value Management Toolkit (PVMT) is used to apply FHA-related properties to model elements. Fields that are typically observed in an FHA table, such as the failure condition and its effects, the associated flight phase, the classification, and the targeted safety requirements, are created in PVMT and applied to each function. The functions are colour-coded according to the evaluated criticality of associated failure conditions, and the architect is able to inspect all failure conditions associated with a particular function.

Thus, the MdFHA method enables an early development of the FHA using a system architecture specification model as the basis. The system specification model can be built as the FHA is conducted, starting from aircraft-level functional decomposition to system-level function specification, followed by the development of a system and logical architecture. The MdFHA process can also be applied to an existing system architecture specification, with the key enabler being the use of Capella diagrams to conduct analysis to inform the FHA. The link between the generic element descriptor and the Capella MBSE tool allows for architectures to be directly converted to a formal MBSE specification model and can provide useful information to support the MdFHA. Overall, the MdFHA approach enables collaborative and discussion-driven development of the FHA between the system architect and the safety analyst. The reader is directed to [150] for a detailed description of the MdFHA process.

3.6.2 Quantitative Safety Assessment

The quantitative safety assessment methods outlined in the ARP4761 are typically used for architecture validation during the System Safety Assessment (SSA). These include Reliability Block Diagram, Dependency Diagram, Fault Tree Analysis and Markov Analysis. Analyses such as the FTA can also be used qualitatively to inspect common cause failures. This is an important characteristic that can be used to support conceptual design.

Incorporating formal qualitative and quantitative safety assessment approaches for systems architecting in aircraft conceptual design is challenging because of conflicting requirements. On one hand, the exploration of a large design space of candidate architectures and the integration of design space exploration within MDAO evaluation workflows requires safety metrics that can be provided to an optimization algorithm. While the rule-based filtering approach provides a binary metric of whether the architecture complies with safety rules, there also needs to be a means of comparing system architectures based on their safety characteristics. On the other hand, the formal quantitative methods listed above have the risk of not offering meaningful insights based on the level of architectural detail available in conceptual design. Even missing one common cause event in a fault tree can result in the failure probability of the top-level event being inaccurate by an order of magnitude. Thus it is important to understand how analyses like the FTA are carried out and the quantitative and qualitative uses of fault trees. Finally, it is also important to consider how these artifacts are used to communicate information about the system architecture, especially if these are to be used early in the design process.

Consider the timeline of fault tree development outlined in the ARP4761, overlaid with the aircraft development timeline, as shown in figure 3.22. Here, the solid black shapes represent the Fault Tree Artifact development in the traditional system architecture safety assessment process.

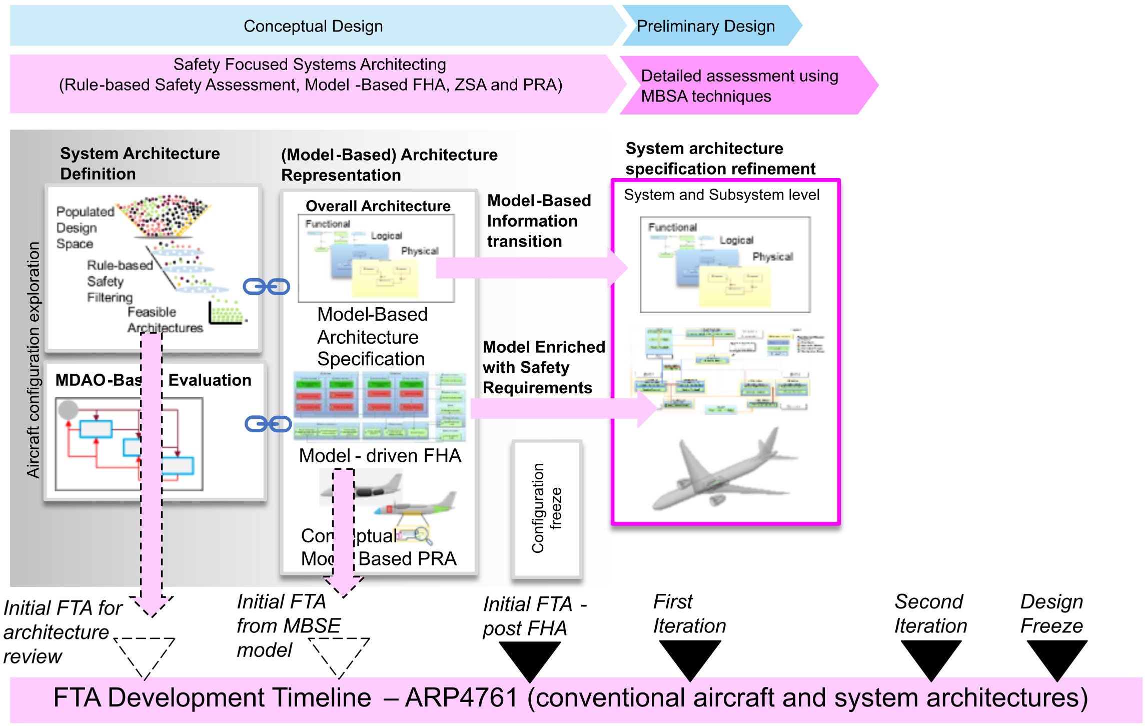

Conceptual Design
Preliminary Design
Safety Focused Systems Architecting
(Rule-based Safety Assessment, Model -Based FHA, ZSA and PRA)
Detailed assessment using MBSA techniques
System Architecture Definition
(Model-Based) Architecture Representation
System architecture specification refinement
System and Subsystem level
Populated Design Space
Rule-based Safety Filtering
Feasible Architectures
Overall Architecture
Functional
Logical
Physical
Model-Based Information transition
Model-Based Architecture Specification
Model Enriched with Safety Requirements
MDAO-Bas Evaluation
Aircraft configuration exploration
Model - driven FHA
Con ptual Mod Based PRA
Configuration freeze
Initial FTA for architecture review
Initial FTA from MBSE model
Initial FTA - post FHA
First Iteration
Second Iteration
Design Freeze
FTA Development Timeline – ARP4761 (conventional aircraft and system architectures)

Figure 3.22: Introduction of FTAs early in the aircraft development timeline

The initial FTA is created after the FHA is performed, and the fault tree goes through several iterations during the different design stages. These iterations are based on architecture reviews between system designers and safety analysts. They rely heavily on information about functional failure conditions and severity classifications from the FHA and also use the system architecture specification as a reference for the architecture. Incorporating quantitative safety analysis, such as the FTA, in conceptual design implies that the development of the initial FTA takes place during system architecture definition or after system architecture representation.

This leads to several considerations. The first is that the FTA may not be usable as a formal architecture validation tool since, at this stage, the emergent safety characteristics of the architecture are not captured because the architecture is not defined at the requisite level of detail for such activities. Second, the FHA has not yet been performed, as the architecture definition stage is more focused on exploring a broad design space instead.

Thus, a useful application of the FTA at this stage is to allow early architecture review to improve collaboration between the system architect and the safety analyst. The proposed safety-focused systems architecting framework introduces the use of the FTA for architecture review right after the system architecture definition and rule-based safety filtering step.

Here, the intent is that once the system architecture design space has been reduced to a tractable size, promising architectures can be reviewed using the FTA qualitatively and quantitatively to down-select system architectures. The added benefit here is that the system architect can already provide the safety analyst with an artifact within the safety domain containing a safety viewpoint of the system architecture, allowing the analyst to provide early feedback. Additionally, depending on how the architect chooses to use this capability, it can also be possible for the architect to specify an architecture and refine it iteratively using the early FTA to provide feedback on the reconfiguration of the system architecture.

A second consideration is that once an architecture is selected and developed in an MBSE environment, the MdFHA can be performed. The specification model now contains additional safety-related information that can be used to model failure conditions using MBSA tools. At this stage, sufficient detail can be provided to the system model by specifying the transfer functions of components and selecting the probability distribution function to model component failures to inform a fault tree at a higher level of detail. This brings the first iteration of the fault tree within the scope of conceptual design, as shown using the clear triangles in figure 3.22.

Finally, when the system architect is configuring the architecture, there are two processes at play. A foreground process is shown in figure 3.23, where the architect selects system architecture components, allocates component redundancies, assigns component interconnections, and develops a representation of the architecture which can be reviewed by system architecture subject matter experts and safety analysts.

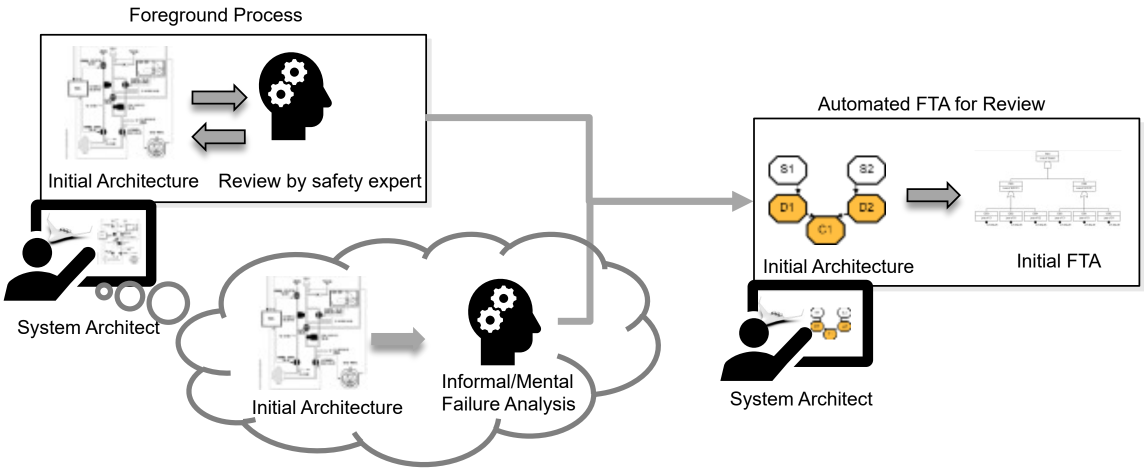
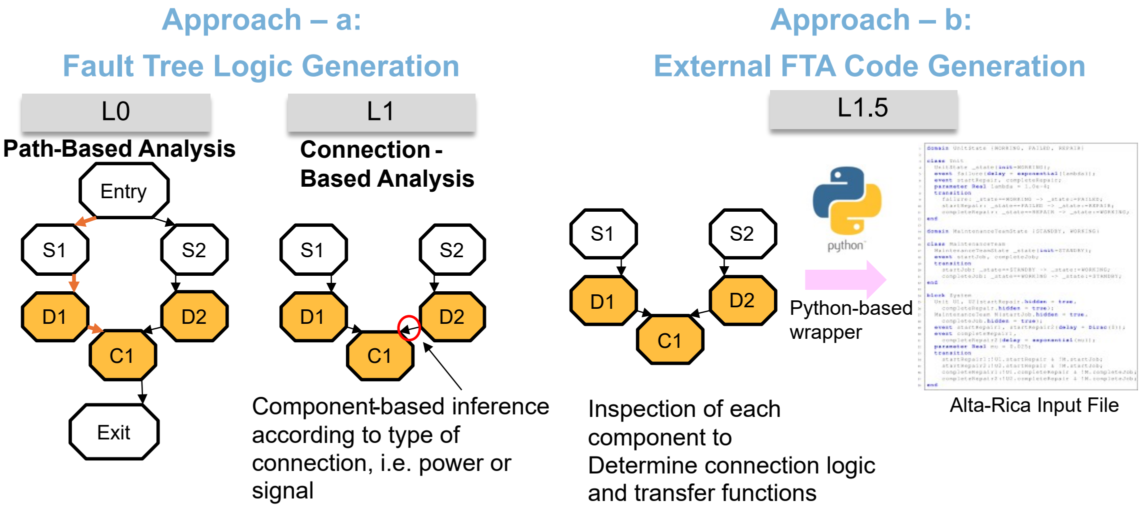


Figure 3.23: Foreground and background processes while architecting

However, while the architect is preparing the architecture, a background process takes place where the architect makes decisions on component redundancies and interconnections based on prior experience or to mitigate specific failure conditions. At this stage, the system architect can benefit from an interactive representation of the architecture that can provide quantitative insight into the changes that the architect makes. Furthermore, having a system architecture representation that can be used to generate safety artifacts, such as the FTA, which can be used for architecture review, can aid collaboration between the systems architecting and safety assessment disciplines.

This section will present an approach to leverage the characteristics of the generic element descriptor to integrate fault tree analysis within the conceptual design phase and also to define the key enablers that allow the transformation of an MBSE model into an MBSA model in the AltaRica 3.0 environment.

This thesis introduces a framework for performing quantitative and qualitative safety assessment using fault tree representation derived from the generic element descriptor. Figure 3.24 shows the different ways in which information from the graph descriptor can be used to generate fault trees at different levels of granularity. The L0 component of approach - a uses the inherent connectivity characteristic of network graphs and graph algorithms that determine the number of paths between any two nodes in the graph. This is termed the path-based approach, and it helps determine the number of power paths through the system that will result in the system being functional.


Approach – a:
Fault Tree Logic Generation
L0
Path-Based Analysis
Entry
S1
S2
D1
D2
C1
Exit
L1
Connection - Based Analysis
S1
S2
D1
D2
C1
Component-based inference according to type of connection, i.e. power or signal
Approach – b:
External FTA Code Generation
L1.5
S1
S2
D1
D2
C1
python
Python-based wrapper
Inspection of each component to Determine connection logic and transfer functions
Alta-Rica Input File

```
domain UnitState {WORKING, FAILED, REPAIR}

class Unit
  UnitState _state(init=WORKING);
  event failure(delay = exponential(lambda));
  event startRepair, completeRepair;
  parameter Real lambda = 1.0e-4;
  transition
    failure: _state==WORKING -> _state:=FAILED;
    startRepair: _state==FAILED -> _state:=REPAIR;
    completeRepair: _state==REPAIR -> _state:=WORKING;
end

domain MaintenanceTeamState {STANDBY, WORKING}

class MaintenanceTeam
  MaintenanceTeamState _state(init=STANDBY);
  event startJob, completeJob;
  transition
    startJob: _state==STANDBY -> _state:=WORKING;
    completeJob: _state==WORKING -> _state:=STANDBY;
end

block System
  Unit U1, U2(startRepair.hidden = true,
    completeRepair.hidden = true);
  MaintenanceTeam M(startJob.hidden = true,
    completeJob.hidden = true);
  event startRepair1, startRepair2(delay = Dirac(0));
  event completeRepair1,
    completeRepair2(delay = exponential(mu));
  parameter Real mu = 0.025;
  transition
    startRepair1:!U1.startRepair & !M.startJob;
    startRepair2:!U2.startRepair & !M.startJob;
    completeRepair1:!U1.completeRepair & !M.completeJob;
    completeRepair2:!U2.completeRepair & !M.completeJob;
end
```

Figure 3.24: Methods of extracting information from the graph descriptor to automatically generate fault trees

The L1 component of approach - a introduces logic into evaluating the paths and determining the causalities that identify whether a component has failed or not. In this case, the simple path approach is used as a starting point, but the fault tree definition takes into account the power type that each component has and whether a component is operational or failed, depending on the available incoming connections from other components. For example, consider figure 3.26 where a consumer element representing an actuator is shown connected to both a hydraulic distribution and a controller element. In this scenario, the actuator will perform its function if both power and control are available. Thus, the connection-based approach examines if the connection logic for each component is satisfied in the descriptor and then writes an FTA based on this logic.

Approach - b incorporates elements of approach a and b, and provides a higher level of fidelity to the generated FTA. It consists of using the generic descriptor and the connection logic for each component and converting it into an AltaRica MBSA model. The fault tree artifact is then generated by AltaRica and visualized in the Arbre Analyst tool [151].

A description of each of the aforementioned methods is provided below:

Approach a) - L0 : Simple path approach

The simple path approach uses the connectivity characteristics of the graph-based generic element descriptor and determines the number of paths through the system that enable the system to perform its function. The paths through the system are determined using the simple path algorithm of [152] implemented in the Networkx python package [133].

A predicate for using this approach is to assign "Entry" and "Exit" nodes to the graph, which represent the starting and ending points of the path algorithm. Figure 3.24 shows an example of the paths that are found between the "Entry" and "Exit" nodes of the system. The algorithm returns a set of paths where each path is represented as a list of node names. Each of these paths is in parallel with the others, meaning that all of the paths would have to be inoperative for the system to be unable to perform its function. Each path consists of components which are in series, and the failure of one component in a path will render that path non-operational.

As the generic element descriptor can also store information within each node, parameters such as failure rate identified from literature or supplier data can be assigned to each node. This enables the computation of the overall failure rate of the system represented using the generic element network graph. Although this computation of system reliability is by no means comprehensive, it is sensitive to redundancies in components and component interconnections as the number of paths through the system changes when additional redundancy or connections are added. Thus, the path-based approach is useful for comparing architectures in a large design space.

In ASSESS L1-M1, the system failure rate computation for a large design space is implemented using the simple path approach, with the overall failure rate being computed as the union of the failure rate values of each path as originally described in [153]. An important assumption for this analysis is that the component failure rates are assumed to be constant, and an exponential distribution is used to model overall reliability. However, the implementation is flexible for the addition of other probability distributions.

The path-based approach enables the estimation of system reliability and overall failure rate. However, in some cases, the architecture design space filtering process will yield a set of architectures with similar estimates of system reliability. In these situations, it is expected that other metrics, such as weight and overall aircraft level integration impact on fuel burn, will be used to select the best candidate architecture. Additional metrics of network graphs can then be used, subject to simplifying assumptions. One such metric is the network density, which is defined as follows:

$$d = \frac{m}{n(n-1)}, \tag{1}$$

where n is the number of nodes and m is the number of edges in G.

Here, the assumption is that, when comparing graphs, the greater the number of edges or component interconnections, the more likely that the overall part count of the system will be higher, thereby resulting in a higher system weight. However, it is to be noted that the density metric is to be used as a second parameter after the overall system failure rate for comparing architectures in cases where the system architectures have the same order of magnitude in their failure rates and already meet minimum safety requirements.

If graph-level metrics such as density are not able to provide additional clarity to choose between equally ranked candidate architectures, then the simple path method enables the consideration of node or component-level metrics. The "betweenness centrality" metric measures the extent to which all available shortest paths pass through a specific node; it is implemented as a function in Networkx [154]. It can help highlight over-reliance on a specific node for redundancy in connections from power sources to power consumers.

Thus, the path-based approach supports the early elicitation of system architecture safety metrics and enables the comparison of architectures in a large design and further down selection. The focus of this approach is to quantify, at the highest level of granularity, the variation in component redundancy. It does not provide any detailed insight, and the failure rate estimation is based on extensive simplification and does not take into account detailed boolean computations that are typically seen in fault trees to estimate top-level failure probabilities. However, the path-based analysis provides the logical groundwork to build simplified fault trees that can be used to review the architecture and examine the sensitivity of component failures on the overall system safety.

Fault Tree development using the path-based approach

Figure 3.25 shows an architecture comprised of two source elements, two distribution elements, and one consumer element. The simple path algorithm provides all the paths through the system. Each path is in parallel, and each component is in series with other components within a specific path. As a result, the loss of a path can be represented as an intermediate event in a fault tree where the loss of each component within a path can be represented under an OR gate. The loss of the entire system is a result of the loss of both paths that are available and therefore represents an AND gate.

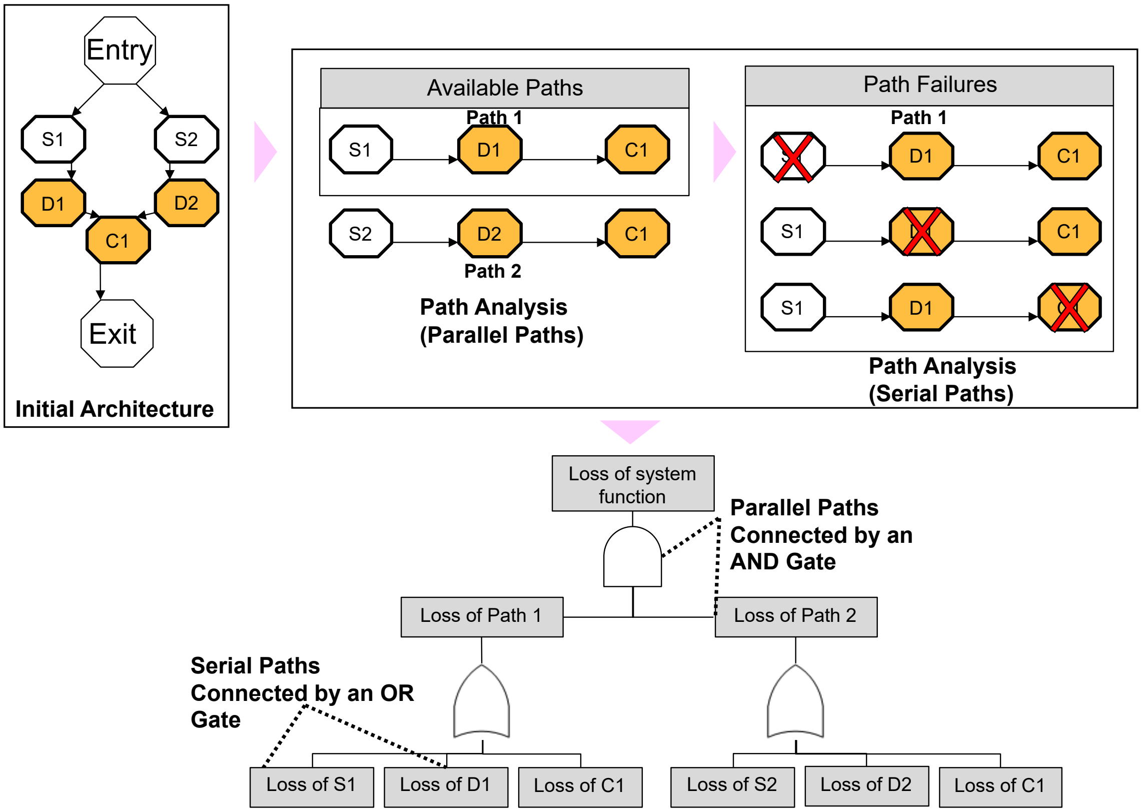


Figure 3.25: Description of paths through a system

The fault tree logic developed from the simple path analysis of the generic element graph descriptor is now converted into a fault tree structure and written in a ".opsa" format[7]. The fault tree is visualized in Arbre Analyst by importing the ".opsa" file, and the top-level event can be evaluated by running the XFTA [155] engine within Arbre Analyst. The information extracted from the generic element descriptor can now be used to build and solve a fault tree. The XFTA computation provides the minimum cutsets and other metrics, such as the failure rate of the top events and the contribution of each event to the loss of the system.

Though the path-based approach to building FTAs is useful for early evaluation and review of the system architecture, it considers only the power system and power allocation view. It is not flexible enough to handle components such as controllers, and it does not take into account different ways in which a component can fail. For example, the path-based approach simply considers power flow through the system but cannot differentiate between the failure of an actuation element due to loss of power or loss of control signal. It serves merely as a means of visualizing allocations and power system component redundancies while using the fault tree representation to identify minimum cutsets and common cause failures. However, the generic element descriptor still contains enough information to derive more detailed logic to inform the development of an improved fault tree. This is done using the connection-based approach.

Connection-based approach

Figure 3.26 shows two system architectures represented using the generic element descriptor. The connection-based approach considers the incoming and outgoing edges of each element. For example, the operation of the hydraulic consumer shown below requires that it be supplied with power. The hydraulic consumer will not be able to perform its function if it is deprived of all power supply, which means that both incoming connections would have to be lost. Until now, this is similar to the path-based approach, which also represents the dependency of the component on power. However, if the second example is considered, an additional element is introduced, representing the chain of control to the hydraulic consumer.

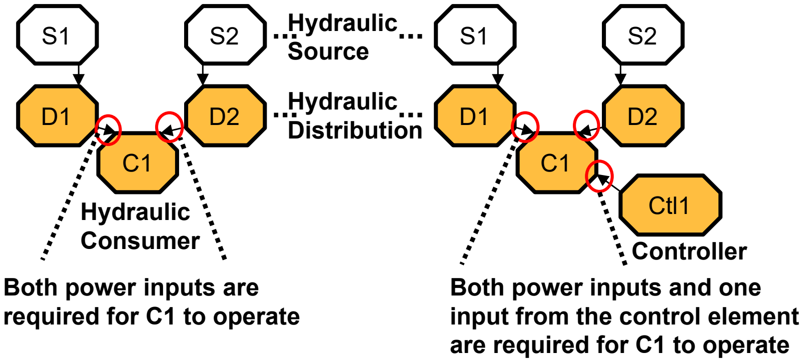

S1
S2
Hydraulic Source
S1
S2
D1
D2
Hydraulic Distribution
D1
D2
C1
Hydraulic Consumer
C1
Ctl1
Controller
Both power inputs are required for C1 to operate
Both power inputs and one input from the control element are required for C1 to operate

Figure 3.26: Example of connection-based logic

The path-based approach would not be able to differentiate between the logic introduced by the control element and those of the other elements. However, using the connection-based approach, it is now possible to include logic that describes that the hydraulic consumer will be operational as long as it receives power and a control input. If either all power is lost or

[7] An example of this format is shown in section B.2 of appendix B.

if all input from the controller is lost, then the hydraulic consumer will be non-operational. This additional information can add detail to the fault tree and lead to more insight into the system architecture.

The implementation of the connection-based approach is at the fault tree logic level; the simple paths would therefore also have to be computed for the connection-based approach in the same manner as for the purely path-based approach. The difference between the two approaches appears while writing the fault tree description in the ".opsa" file. The process for writing this file for both path-based and connection-based approaches is described below.

Writing a fault tree to a ".opsa" format

The ".opsa" file format is part of the Open-PSA model exchange framework. It provides a means of defining fault tree logic and specifying parameters associated with the failure rates of individual components. The structure of the file is shown below in figure 3.27. The parameters section allows the definition of individual parameters that can be assigned to different top-level events or used to store an internal failure rate.

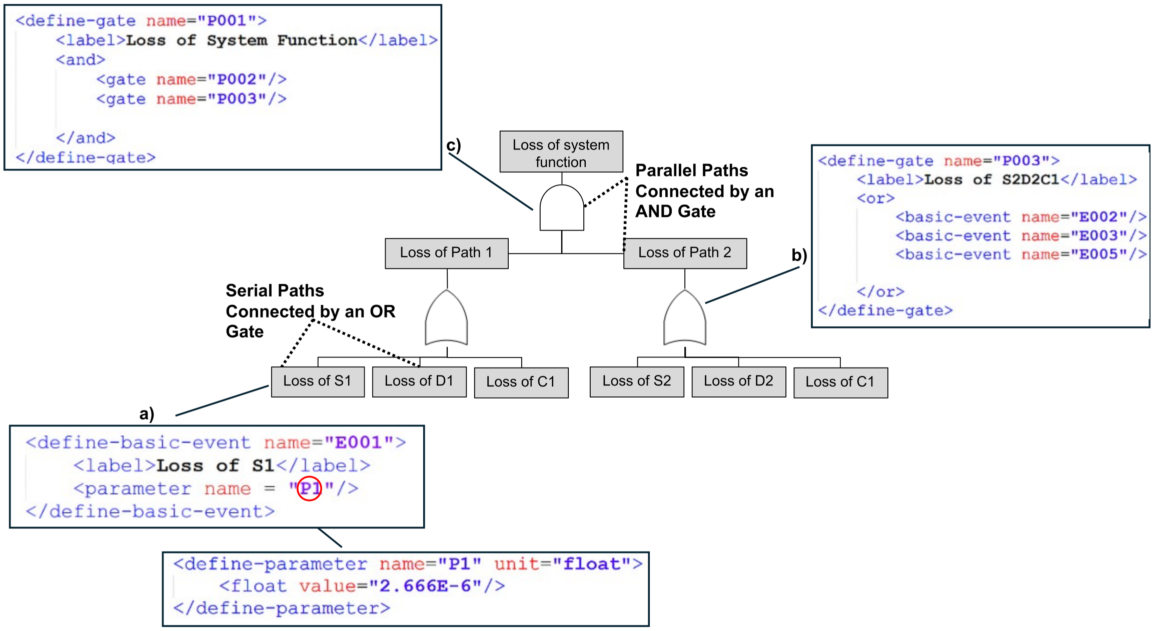
Each basic event in the fault tree is defined in a specific tag. Intermediate events are specified using a name and are also assigned the gate that contains the basic events that lead to that particular intermediate event. The top event is defined in the same way. The identifier of the parameter assigned to each basic event is included in the definition of each basic event. Consider the notional architecture shown in figure 3.25. The path analysis reveals several potential power paths through the system. These can be converted into fault trees as follows:

1. All the parallel paths are identified and listed

2. Basic events are defined as the loss of individual components due to internal failures, and the failure rates are specified using the "<define-parameter>" tag written to the definition of the basic event in the .opsa file as seen under a) of figure 3.27

3. Basic events are related to intermediate events using gate definitions as shown in part b) of figure 3.27

4. Intermediate events are then related to the top-level "Loss of system function" event using a gate definition in part c) of figure 3.27



```
<define-gate name="P001">
    <label>Loss of System Function</label>
    <and>
        <gate name="P002"/>
        <gate name="P003"/>

    </and>
</define-gate>
```

```
<define-gate name="P003">
    <label>Loss of S2D2C1</label>
    <or>
        <basic-event name="E002"/>
        <basic-event name="E003"/>
        <basic-event name="E005"/>

    </or>
</define-gate>
```

```
<define-basic-event name="E001">
    <label>Loss of S1</label>
    <parameter name = "P1"/>
</define-basic-event>
```

```
<define-parameter name="P1" unit="float">
    <float value="2.666E-6"/>
</define-parameter>
```

Figure 3.27: Example of writing a .opsa fault tree using the path-based approach

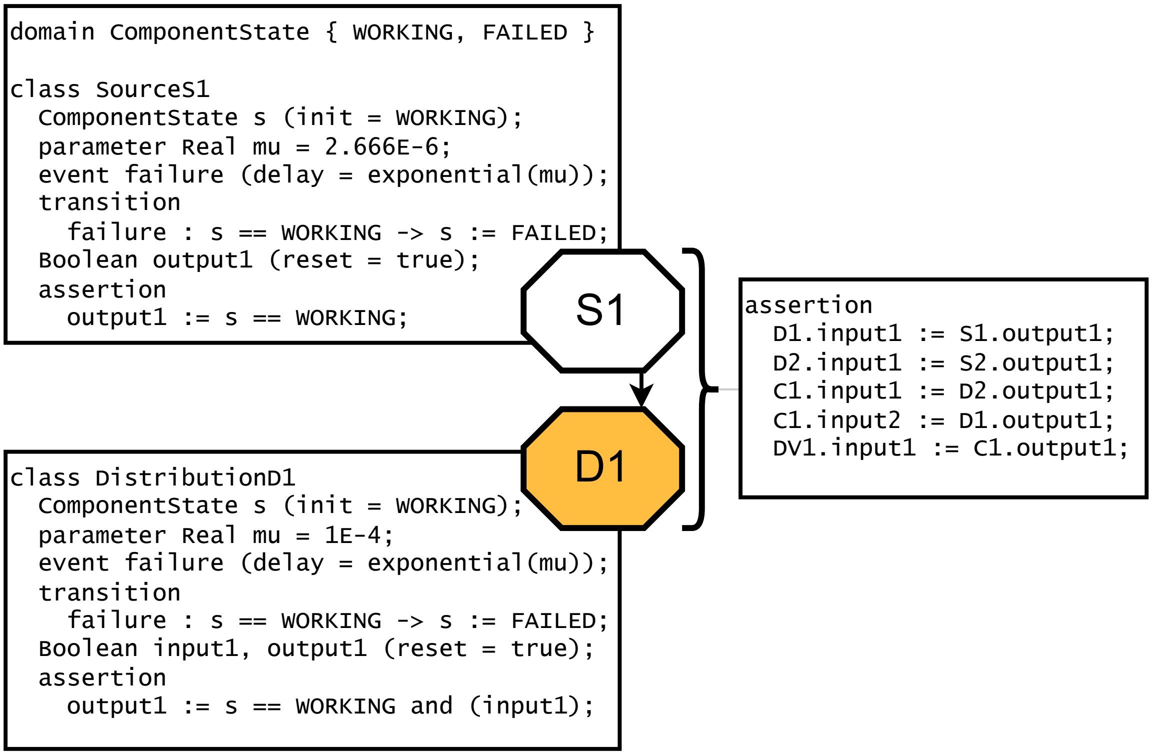
This process is automated in ASSESS L0, with the only inputs that are required being the generic element descriptor with component failure rates associated with each element.

Graph to AltaRica 3.0

The approach to developing fault trees from the graph-based generic element descriptor presented in the previous section focused on extracting information about the power paths through the system that enables the system to remain operational. However, this does not constitute a robust description of the system or its safety characteristics, i.e., it cannot be considered a safety model. Nevertheless, the graph at this stage can still contain information such as the logic that drives the state of each component in the architecture, and the component information could be used to inform a preliminary safety model.

The process of building a formal safety model from the graph proposed in this thesis takes the graph-based generic element description of a system architecture, extracts the connection logic and failure rate information assigned to each component, and converts it into a simple safety model in the AltaRica 3.0 environment. This automated definition of a safety model in an MBSA environment is used to investigate important failure conditions and to develop an initial fault tree with more detail than those generated from the path-based or connection-based approaches.

The graph-to-AltraRica process is implemented by developing a component-to-component mapping template in AltaRica that enables each component in the graph to be specified in the AltaRica input script. Here, a standard template is filled with information extracted from each component in the model. The inputs and outputs of each component are examined; based on the type of component being considered, the necessary modifications to the input-output definition of the AltaRica model are made. A description of this process is included below, and the syntax is described in figure 3.28 for the example of the notional architecture shown in figure 3.25.



```
domain ComponentState { WORKING, FAILED }

class SourceS1
  ComponentState s (init = WORKING);
  parameter Real mu = 2.666E-6;
  event failure (delay = exponential(mu));
  transition
    failure : s == WORKING -> s := FAILED;
  Boolean output1 (reset = true);
  assertion
    output1 := s == WORKING;
```

```
class DistributionD1
  ComponentState s (init = WORKING);
  parameter Real mu = 1E-4;
  event failure (delay = exponential(mu));
  transition
    failure : s == WORKING -> s := FAILED;
  Boolean input1, output1 (reset = true);
  assertion
    output1 := s == WORKING and (input1);
```

```
assertion
  D1.input1 := S1.output1;
  D2.input1 := S2.output1;
  C1.input1 := D2.output1;
  C1.input2 := D1.output1;
  DV1.input1 := C1.output1;
```

Figure 3.28: Example of AltaRica syntax developed from a generic element descriptor

A typical safety model in AltaRica is declared using syntax that consists of several elements representing information about a component, a class of components, component instances, a system, and the failure conditions that need to be evaluated. The class syntax enables the declaration of component classes which can store properties about each component. An important property that needs to be stored is the failure rate, along with the failure model that is used, for example, exponential, Weibull, etc. Presently, only an exponential distribution model for component failures is considered, as it is sufficient to demonstrate the utility of the graph-based descriptor and the present method for generating simplified quantitative safety metrics. The integration of additional models is considered trivial and not covered in this thesis.

Within the declaration of a component class, other properties of the component must be specified. These include the state of the component and the various states into which it can transition. Here, under the transition section of the class definition, a failure can be declared by specifying the state "s" as going from "WORKING" to "FAILED." The output statements allow the declaration of different outputs, which will be used in other sections of the AltaRica model. The assertion statement defines under which state the component's output is available.

The "block" statement allows the declaration of components as instances of the component classes. Thus, here, SourceS1 refers to a component S1 that is defined using the class SourceS1. The class names are concatenated to ensure ease of automation in generating the AltaRica syntax from the generic element descriptor.

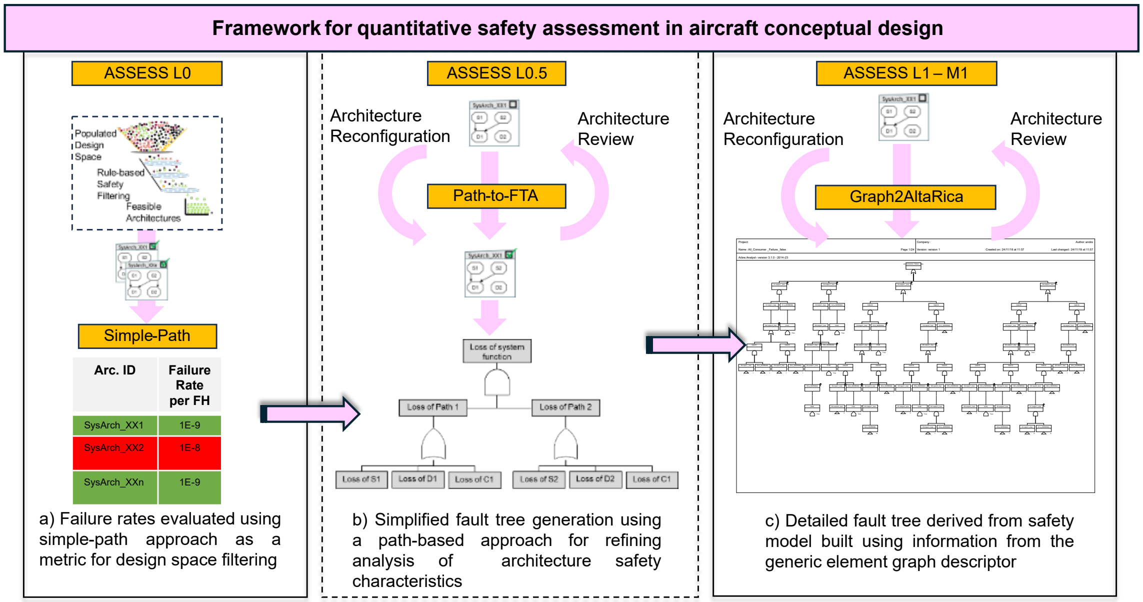
The definition of observers requires manual input. However, since the methods presented in this thesis are tailored to conceptual design for power system architectures, three important failure conditions are predefined and written to the AltaRica model. These are as follows :

1. All engine failure
2. Single engine and opposite distribution (electrical or hydraulic) failure
3. All actuator or end consumer failure

These are generated automatically using a script that reads the descriptor and identifies which elements represent the power generated by the engines, the distribution systems, and the actuators or devices based on the information contained in the graph nodes. Finally, the assertion section, which outlines how component inputs and outputs are connected, is generated using the connection logic from the generic element descriptor.

Figure 3.28 shows an example of how the assertions are generated. Here, a source is connected to a distribution element. The outgoing edge of the source element is identified and traced to the input of the distribution. The output of the sources is labelled as "output," followed by a number in sequential order of the outgoing connections. When compiling the AltaRica model, the list of outputs is parsed for each node in the generic element descriptor and traced to the node that receives the connection. In this way, for each node, the inputs and outputs can be compiled into a single list to be declared under the assertions section for each component. The complete syntax of the AltaRica safety model for the notional architecture shown in 3.25 is included in appendix B.2.

Finally, having described the various methods of extracting architecture information from the graph-descriptor and performing quantitative safety assessment at different levels of fidelity, it is possible to integrate these methods into the ASSESS framework, as shown below in figure 3.29. The notation for the levels of detail provided here is kept consistent with those associated with the various ASSESS modules shown in figure 3.2. However, there is a difference in notation from that which is used in figure 3.24, as the focus there was on describing different levels of fidelity of information extraction and transfer for safety assessment, whereas this section focuses on showing how the tools developed using the aforementioned information extraction and transfer processes at various levels of fidelity are integrated within various ASSESS modules.



a) Failure rates evaluated using simple-path approach as a metric for design space filtering

b) Simplified fault tree generation using a path-based approach for refining analysis of architecture safety characteristics

c) Detailed fault tree derived from safety model built using information from the generic element graph descriptor

Figure 3.29: Mapping between different conceptual quantitative safety assessment methods and various ASSESS modules

The first method implemented within ASSESS L0 is the simple path-based approach. One of several ways in which this is envisioned is to filter a large design space of candidate architectures to determine the overall system failure rate and then down-select architectures that meet the safety requirements. This is shown in figure 3.29 a). In addition to the simple path approach, the path to FTA method shown in figure 3.29 b) is integrated in L0 but is considered to be more detailed and therefore part of an L0.5[8] level of detail. Here, the intent is to facilitate an improved estimation of the failure rate for the complete loss of system function, taking into account exclusively the power view, the control view or the fuel flow view. Another application of the early FTA generation in L0.5 is to facilitate fast architecture review and reconfiguration. The system architect can define an architecture in the generic element representation and generate a fault tree without having to provide additional information required to build an AltaRica safety model, as in ASSESS L1-M1.

The quantitative safety assessment module in ASSESS L1-M1 uses the graph descriptor, properties stored in the graph descriptor, and the connections between elements to build an AltaRica safety model, which is then used to build a fault tree. As shown in figure 3.29 c), this capability can be used to evaluate different failure conditions and to review and reconfigure the architecture based on the FTA results. Moreover, an architecture selected at the L0 level, evaluated further in the L0.5 level, can be studied further and configured using this capability of the L1 module. Section 3.6.3 below shows several case studies that demonstrate the use of the modules described above.

3.6.3 Application: Quantitative Safety Assessment in Conceptual Design

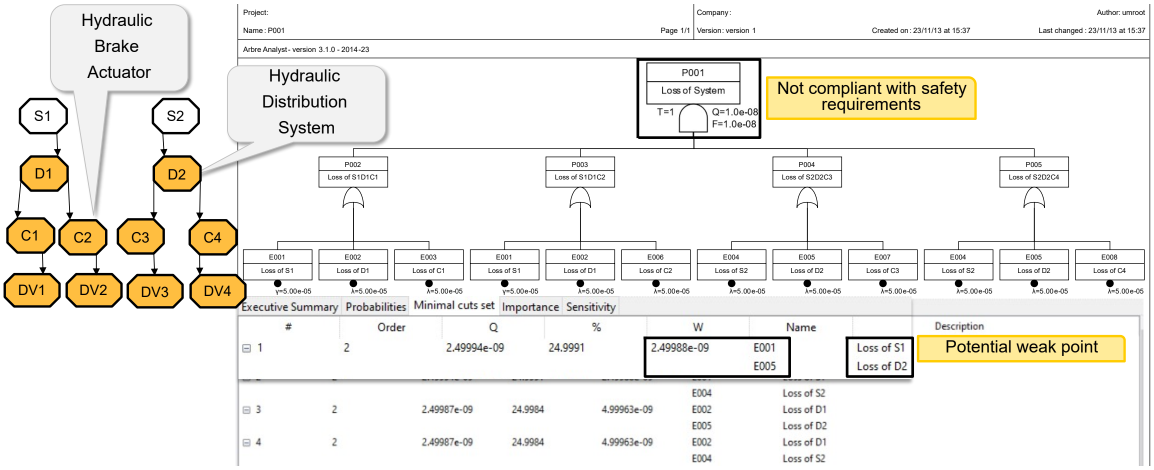
This section presents three studies that demonstrate the utility of the graph-based quantitative safety methods described in section 3.6.2. These case studies focus on the

[8] This is still a part of ASSESS L0 although it is assigned L0.5 to highlight a slight increase in the level of detail in the assessment.

aircraft landing gear braking system and power and control architecture and are intended to demonstrate the early analysis and architecture review that a system architect can perform while specifying the system architecture during aircraft conceptual design.

Aircraft landing gear braking system - path-based FTA

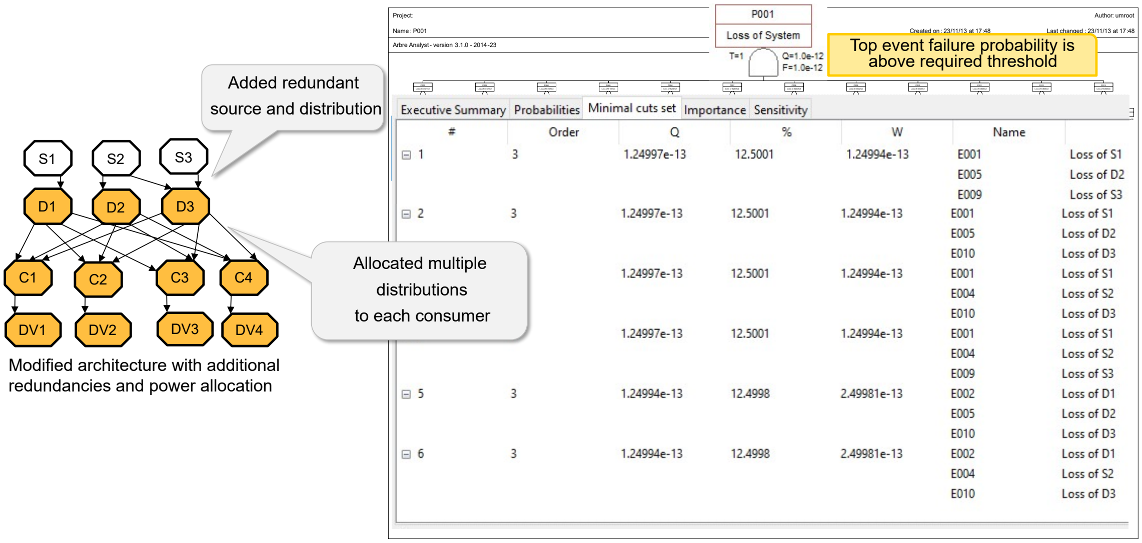
Consider a system architecture specialist within the advanced design department of an aircraft manufacturer tasked with outlining the landing gear braking system of a new aircraft program. The architect needs to specify the technology, the components, and the allocation of power generation sources to distribution components and further to power consumption components. Instead of relying on past architectures, the system architect can use a combination of safety rules and design experience to configure an initial draft system architecture as shown in the left portion of figure 3.30. Here, the system architecture is comprised of four hydraulic brake actuation units, two hydraulic distributions and two sources of hydraulic power. The architect then requires a quick evaluation of the safety characteristics of these architectures to determine if there are sufficient redundancies in power distribution to ensure compliance with quantitative safety requirements. The path-based FTA enables a fast evaluation without the architect having to manually define the fault trees. It helps the architect understand the impact of redundancies on the overall failure rate of the system. In figure 3.30, the initial architecture is shown to not meet the safety requirement for the loss of the braking function. Using XFTA and the Arbre Analyst visualization tool, the architect can also see potential weak points in the system by inspecting the minimal cut sets. Here the architect sees that the loss of S1 and opposite distribution D2 can lead to the loss of the braking function and that the failure rate for this event is unacceptable.



#	Order	Q	%	W	Name		Description
⊟ 1	2	2.49994e-09	24.9991	2.49988e-09	E001 E005	Loss of S1 Loss of D2	Potential weak point
					E004	Loss of S2	
⊟ 3	2	2.49987e-09	24.9984	4.99963e-09	E002 E005	Loss of D1 Loss of D2	
⊟ 4	2	2.49987e-09	24.9984	4.99963e-09	E002 E004	Loss of D1 Loss of S2	

Figure 3.30: Initial review of draft architecture using path-based FTA

Based on this rapid analysis, the architect adds a third redundancy (D3 and S3) in the hydraulic distribution system and an additional source of hydraulic power, as shown in figure 3.31. The architect then reruns the analysis and inspects the resulting fault tree and minimum cut set list, which shows that the failure rate of the top event meets the safety requirements. Also, on examining the minimum cut set list, it is evident that the failure rate of the top cut set is low and exceeds the requirement by several orders of magnitude. Finally, the architect can also examine the impact of the technology of different components on the overall safety characteristics of the system by modifying the individual component failure rates.


Added redundant source and distribution
S1
S2
S3
D1
D2
D3
C1
C2
C3
C4
DV1
DV2
DV3
DV4
Modified architecture with additional redundancies and power allocation
Allocated multiple distributions to each consumer
Top event failure probability is above required threshold
Executive Summary
Probabilities
Minimal cuts set
Importance
Sensitivity

Project:
Name : P001
Arbre Analyst - version 3.1.0 - 2014-23
P001
Loss of System
T=1
Q=1.0e-12
F=1.0e-12
Author: umroot
Created on : 23/11/13 at 17:48
Last changed : 23/11/13 at 17:48

#	Order	Q	%	W	Name	
1	3	1.24997e-13	12.5001	1.24994e-13	E001 E005 E009	Loss of S1 Loss of D2 Loss of S3
2	3	1.24997e-13	12.5001	1.24994e-13	E001 E005 E010	Loss of S1 Loss of D2 Loss of D3
		1.24997e-13	12.5001	1.24994e-13	E001 E004 E010	Loss of S1 Loss of S2 Loss of D3
		1.24997e-13	12.5001	1.24994e-13	E001 E004 E009	Loss of S1 Loss of S2 Loss of S3
5	3	1.24994e-13	12.4998	2.49981e-13	E002 E005 E010	Loss of D1 Loss of D2 Loss of D3
6	3	1.24994e-13	12.4998	2.49981e-13	E002 E004 E010	Loss of D1 Loss of S2 Loss of D3

Figure 3.31: Assessment of modified architecture with additional redundancies and power allocations

The path to FTA approach enables a quick evaluation of the safety characteristics of the system architecture. However, it neglects logic that is inherent in the connections between components in the system architecture. The system architect may also have to account for different types of components and the interaction of signal and power at a component interface. Thus, while in the conceptual design stage, the system architect can resort to developing an FTA using an additional level of fidelity afforded by an equivalent safety model. This is achieved by using the connection-based analysis to identify logic encapsulated in the graph descriptor and the Graph to AltaRica tool to translate the graph descriptor model to an AltaRica safety model.

Graph to AltaRica

The graph to AltaRica approach outlined in section 3.6.2 is implemented in a python-based tool within ASSESS L1-M1 called "graph2AltaRica". An initial assessment of the draft architecture reveals that the all-actuator failure case results in a failure rate for the loss of braking function of $1.05 * 10^{-8}$ failures per flight hour for the same exposure time (1 hr) as in the path to FTA case. The other failure cases that were evaluated showed acceptable failure rates for the loss of braking function. When the architecture was evaluated again after making the modifications shown in 3.31, the overall failure rate was $1.05 * 10^{-8}$, which is of the same order of magnitude as obtained from the path to FTA assessment. This shows that the path to FTA assessment quantifies the worst-case failure scenario, subject to simplifying assumptions such as the lack of consideration of latent failures and only considering the power system architecture. Comparing both assessments, the path to FTA approach emerges as a more suitable choice for quick quantitative assessments of system safety early in the design process for architecture review, while the graph2FTA approach is more suited for interactive architecting and architecture review.

3.7 Integrating MBSE Specification in ASSESS-L1 with an MDAO Workflow

This section describes how a system architecture specification model can be used to provide system architecture input to an MDAO workflow using a demonstrative case study by Jeyaraj et al. from [107]. Figure 3.32 a) shows how the MBSE specification model is linked to a system-level workflow specification that is then used to build a formal MDAO workflow specification. The formal workflow specification is built based on a CPACS description of both the aircraft and system-level tools and their inputs and outputs. This step is performed on the AGILE 4.0 OCE platform, where individual system disciplines and variable names can be defined, and a combined CPACS file can be generated.

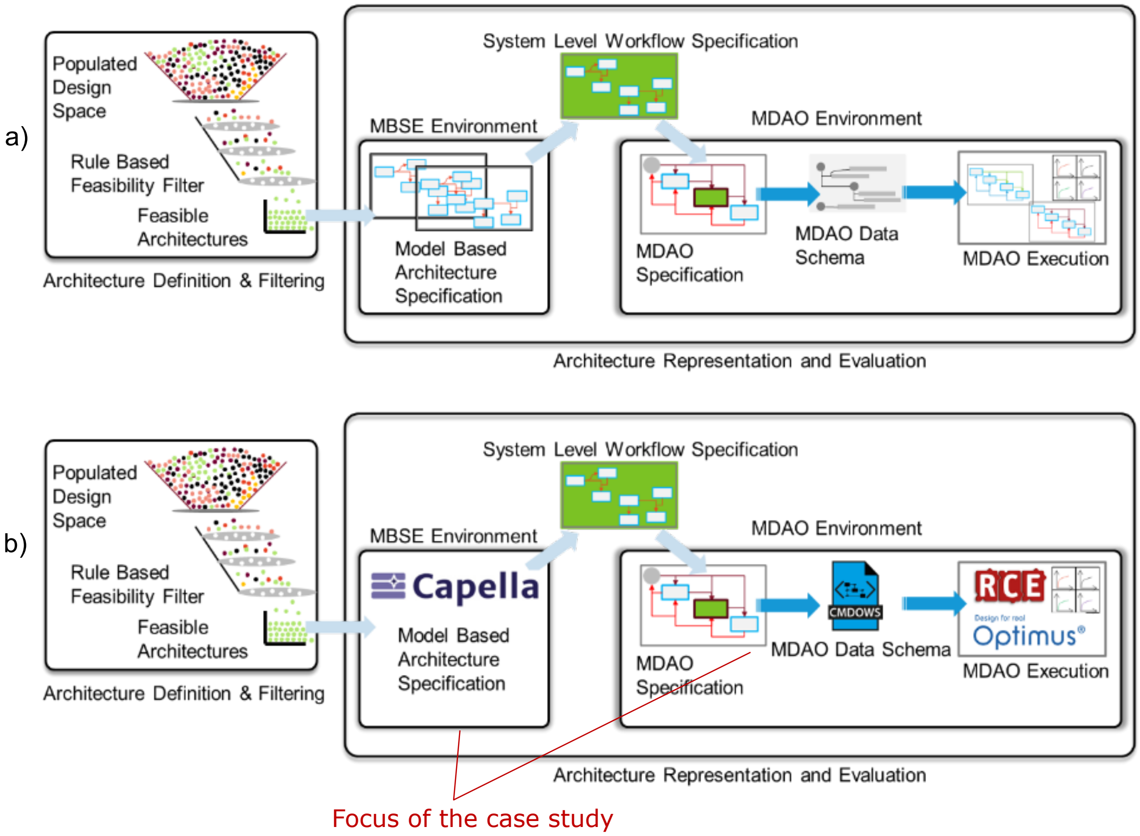

a)
Populated Design Space
Rule Based Feasibility Filter
Feasible Architectures
Architecture Definition & Filtering
System Level Workflow Specification
MBSE Environment
MDAO Environment
Model Based Architecture Specification
MDAO Specification
MDAO Data Schema
MDAO Execution
Architecture Representation and Evaluation
b)
Populated Design Space
Rule Based Feasibility Filter
Feasible Architectures
Architecture Definition & Filtering
System Level Workflow Specification
MBSE Environment
Capella
Model Based Architecture Specification
MDAO Environment
CMDOWS
RCE
Optimus
MDAO Specification
MDAO Data Schema
MDAO Execution
Architecture Representation and Evaluation
Focus of the case study

Figure 3.32: Approach and tool description for MBSE integration with MDAO

Thus, to provide the system architecture inputs to the MDAO workflow through the workflow specification, the inputs only need to be written to the CPACS file. The key step in this process is adding the inputs to elements of the MBSE specification model and then extracting these inputs. In the example presented below, the Capella MBSE tool is used to model the architecture, as shown in figure 3.32 b), and the Property Value Management Toolkit (PVMT) add-on is applied to store system architecture input variables in model elements. A similar approach using the PVMT add-on to integrate the ASSESS L2-M2 tool with the Capella MBSE tool is shown in section C.1 of appendix C.

3.7.1 Application: Flight Control Actuation System Architecture Input to a Collaborative MDAO Workflow

A method for extracting the information stored in the Capella model by parsing the files that store all the model information is described. The property information is then written to the CPACS file and uploaded to the AGILE 4.0 Operational Collaborative Environment (OCE) platform [156], which can then be used to build and execute the formal AGILE 4.0 collaborative workflow. The scope of the proposed method and the example described here is limited to storing and extracting information from the Capella model, as this is the key enabler for the entire MDSE-MDAO link. It is to be noted that when this method was developed and published, no existing tools were available to automatically extract the required information from the Capella model. The recently released pycapellambse [157] package developed by DB InfraGO AG enables interaction with the Capella model by reverse engineering the ".aird" or ".capella" file that stores the model, thereby validating the principle behind the approach presented here.

The exemplar system used to demonstrate the proposed methodology to link an MBSE specification to the collaborative MDAO workflow is the flight control system architecture. Here the key steps that need to be performed are as follows :

1. Enriching the specification model with system-level tool input parameters
2. Extraction of parameters from MBSE (Capella) data model
3. Integration of extracted parameters into an MDAO workflow

The Capella tool is used for the MBSE portion. In Capella, four different levels of abstraction (the operational level, system level, logical level, and physical level) are available to specify a system. Here, the physical level is selected to establish the interface to the MDAO workflow. The link could potentially be established at other specification levels; however, this consideration is outside the scope of this thesis.

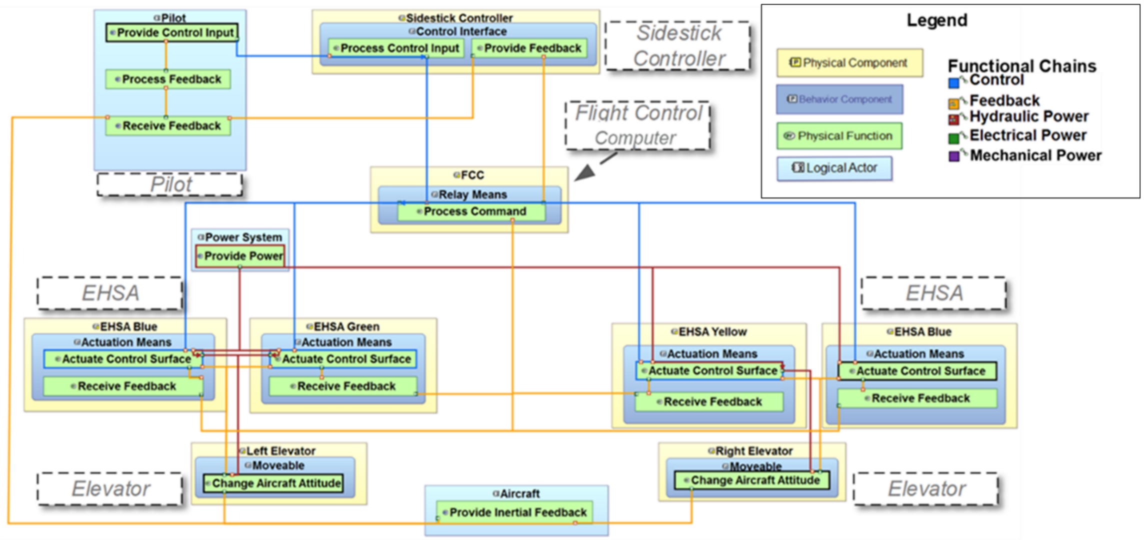


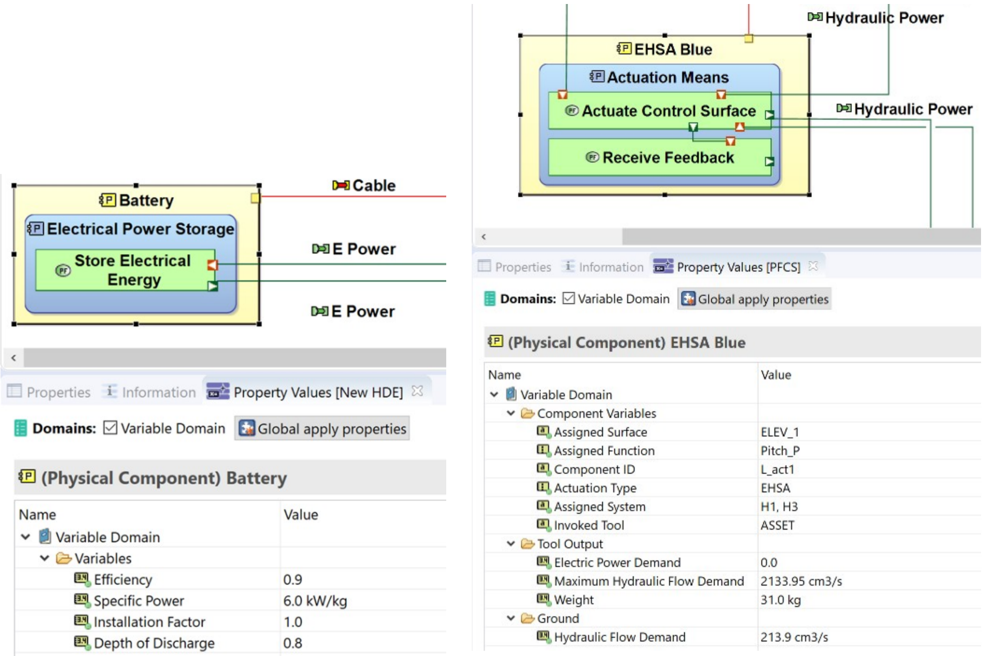
Figure 3.33: Physical architecture model in Capella for a portion of a flight control system (pitch control system), from [32]

Figure 3.33 shows the main physical components, the logical components, and the associated functions of a portion of the flight control system architecture. Functional exchanges are depicted, and the physical links between components are also specified. Here, the key components are the hydraulic actuators, control surfaces, and elements representing the supply of secondary power to each actuator. The flows of signal and power are highlighted using the "functional chain" feature in Capella, wherein individual functional exchanges can be selected and converted into a continuous chain. In this model, all the actuators are of the Electrohydraulic Servo Actuator (EHSA) type, of which two are assigned to each elevator control surface. The type of actuator and the allocation per control surface are important for the sizing of actuators. Each of these elements has specific properties that are used by the system-level sizing and performance workflows to estimate mass, power demand, and other metrics using empirical or physics-based methods.

However, the extraction of information is challenging as most of this information (such as the number and type of actuator) can be inferred from the diagram and other information, which can include hydraulic pressure, supplied power, and component efficiencies, has to be extracted from the underlying data model. This aspect will be covered in the following sections.

Enriching the specification model with system-level tool input parameters

An architecture specification developed in Capella already supports additional information such as component descriptions, architecture summaries, model developmental status, and model validation information. Capella does not natively support the addition of custom or model-specific properties and the interrogation of model elements to access these properties. However, an open-source plugin for Capella called the Property Value Management Tool (PVMT) allows the definition of properties of different data types (string, float, integer, and others), as well as the implementation of specific rules or conditions under which the properties are applied. Furthermore, properties can be made applicable to specific modelling levels, such as the system or physical level. They can also be set to specific modelling artifacts, such as exchanges or functions.



(Physical Component) Battery

Name	Value
Variable Domain	
Variables	
Efficiency	0.9
Specific Power	6.0 kW/kg
Installation Factor	1.0
Depth of Discharge	0.8

(Physical Component) EHSA Blue

Name	Value
Variable Domain	
Component Variables	
Assigned Surface	ELEV_1
Assigned Function	Pitch_P
Component ID	L_act1
Actuation Type	EHSA
Assigned System	H1, H3
Invoked Tool	ASSET
Tool Output	
Electric Power Demand	0.0
Maximum Hydraulic Flow Demand	2133.95 cm3/s
Weight	31.0 kg
Ground	
Hydraulic Flow Demand	213.9 cm3/s

(a) Efficiency, specific power, installation factor, and depth of discharge applied to a physical component in Capella using the PVMT add-on

(b) Input and output parameters required by a system sizing tool applied as properties associated to a physical component in Capella using the PVMT add-on

Figure 3.34: Allocation of properties to physical components in Capella using the PVMT add-on

As an example, figure 3.34(a) shows properties such as efficiency, specific power, installation factor, and depth of discharge applied to a physical component representing a battery. These properties can be used to provide input to a battery sizing tool that will provide the battery's overall weight. Figure 3.34(b) shows properties that are inputs to an actuator sizing tool, such as actuation type and assigned surface. Tool outputs, such as the weight and flow demand, can be categorized separately and specified within the Capella workbench. Properties can also be directly accessed from the underlying data model in Capella. This aspect is explored in the following section.

Extraction of parameters from the Capella data model

Capella is built on the Eclipse Modelling Framework and thus supports customization through modifying the core functionality and developing add-ons [158]. The underlying framework consists of a structured data model which stores all information. The Capella model can be interrogated in the following ways:

1. Using the workbench and diagram editors for visual inspection
2. Accessing the underlying data model and schema[9]

Visual inspection of the Capella model is achieved by the system architect manually interacting with the diagram editor and accessing the PVMT viewpoint tab. The system

[9] Querying the model from within the workbench user interface is also considered here.

architect will then have to transcribe the information manually to a spreadsheet, to an intermediate data format, or directly to the input file of a system-level workflow or tool.

Reading and writing to and from the underlying data model is a practical solution to integrating an MBSE specification in Capella with external tools. This data model is an XML schema that contains all the information about the system architecture specification, including the names of functions, logical components, physical components, functional exchanges, component exchanges, physical links and owned components (functions, logical and physical components), as well as information on assigned properties and their values. Furthermore, the location of graphical elements and changes made to the model are also stored within this schema.

The data model consists of two files to which all the information is written every time a user saves changes to the model. These are the ".melodymodeller" – which is now called ".capella" as of version 5.0.0 and the ".aird" files. The ".aird" file contains information on active diagrams within a Capella project, including the location and formatting of elements in a diagram and a record of changes made to the model. The ".capella" file stores information about each model element, including characteristics, assignments, links, input and output ports, and any assigned property values.

Each model artifact is assigned a unique alphanumeric sequence as an identifier within the schema, and links between elements, such as functional chains and physical links, reference these identifiers in specifying the source and target of each exchange. Although the schema is well structured, the size and tag names can make navigation cumbersome. In order to simplify the traversal of different tags for property extraction and the automation of the process, a simplified schema is created, which contains only the tags that are accessed to extract model element properties. The mapping from the actual schema tags to a simplified version is shown in table 3.5, supplemented by an illustration in the Appendix.

Table 3.5: Mapping between Capella schema and simplified schema tag names

Capella Schema Tag Name	**Simplified Schema Tag Name**
Root:org.polarsys.capella.core.data.capellamodeller:Project	Project
Root:ownedPropertyValuePkgs	Property Value Packages
Child:ownedPropertyValueGroups	Property Value Groups
Child:ownedPropertyValues	Property Values
Root:ownedModelRoots	Models
Child:ownedArchitectures	Architectures
Child:ownedPhysicalComponentPackage	Physical Architecture
Child:ownedPhysicalComponents	Components
Child:ownedPropertyValueGroups	Property Value Groups
Child:ownedPropertyValues	Property Values

An algorithm for extracting model properties from the ".capella" file is explained below in figure 3.35. This process is based on the following assumptions:

1. A predefined list of components is created in Capella, each with its name and unique identifier.

2. A list of properties is defined with the PVMT tool, and variable names are included in the "key" field within the PVMT editor.

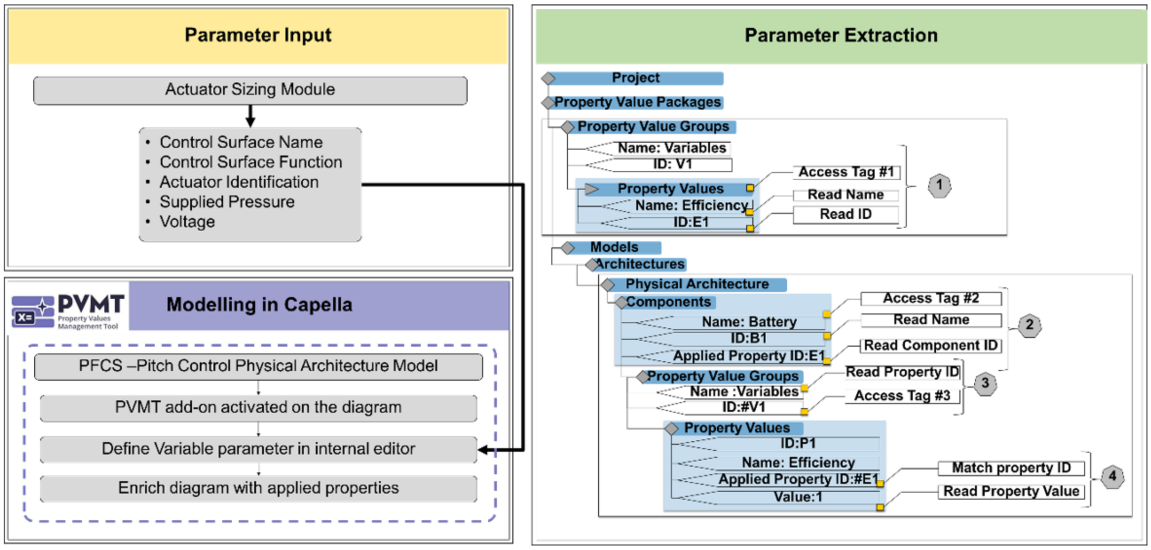


Figure 3.35: Process for parameter input and extraction in Capella, from [107]

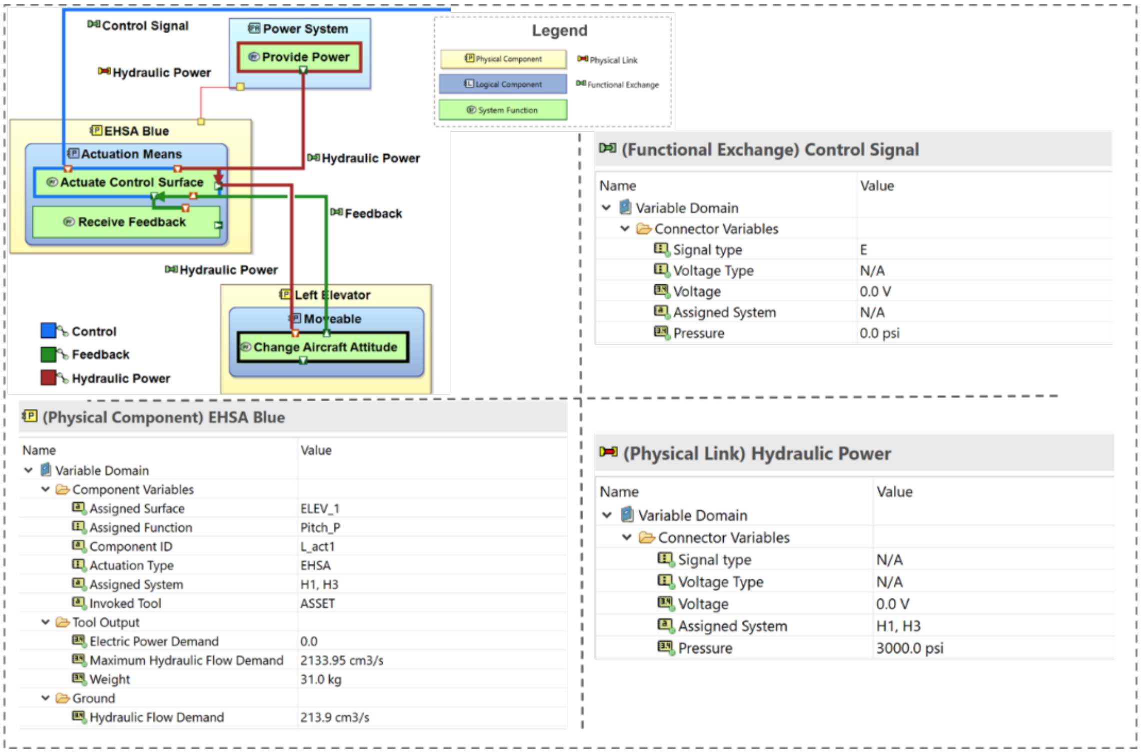
First, the inputs are compiled and added to the Capella model through the PVMT tool. To do so, one needs to activate the PVMT viewpoint in Capella and define and name each property's data type. Once the properties are defined, values are assigned to each field; the model must then be saved. One can extract parameters by first reading the top-level property value group tag and then reading the name and ID of the child property value tag. Second, one can access the components tag and parse for the name, ID, and applied property ID of each physical component. The child property value groups tag is then accessed, and the ID value is read. Finally, the property values tag, which itself is a child of the property value groups tag assigned to the physical component, is accessed. Here, a match is made between the applied property ID and the property value ID defined in the parameter definition step shown in figure 3.35, and then the value of the property is stored.

In some cases, when the number of inputs from the system architecture to a system-level tool or workflow is relatively high, it may be cumbersome to complete an exhaustive specification of properties in Capella. In such cases, parsing the Capella schema in the manner described above can be aided by adding inferential logic from the system architecture specification to define the additional tool input values. This functionality is demonstrated for the system architecture shown below, when linking it to an actuator sizing tool that is part of a system-level workflow. The inputs and outputs of this tool are defined in table 3.6.

In figure 3.36, the physical links from the power system element to the actuator elements are shown to have assigned properties that include power type and pressure. When reading these parameters, one can deduce that the actuator connected to this power system type uses hydraulic power. Similarly, one can assign a parameter called "signal type" to the input signal that is represented by a functional exchange. In this case, the system architect can deduce that the actuator is electrically controlled as the control signal is of type "electric." Combining the logic of these two inferences allows for the actuator to be hydraulically powered and electrically controlled. This results in the identification of the actuator type, and iterating through all the actuator components allows the total number of actuators of a given type to be determined.

Table 3.6: Process for parameter input and extraction in Capella

Required Tool Inputs	Source of Value	Referenced Capella Element
Control Surface name	Specified	Physical Component Property
Control Surface function	Specified	Physical Component Property
Pressure	Specified	Physical Link Property
Voltage	Specified	Physical Component/Link Property
Voltage type	Specified	Physical Component/Link Property
Technology	Specified/Inferred	Physical Component Property
Flight critical system designation	Specified	Physical Component Property
Association to System	Inferred	Physical Component Property
No. of Hydraulic Actuators	Inferred	Physical Component Property
No. of Electric Actuators	Inferred	Physical Component Property
Surface to Actuator Assignment	Inferred	Physical Component Property



(Functional Exchange) Control Signal

Name	Value
Variable Domain	
Connector Variables	
Signal type	E
Voltage Type	N/A
Voltage	0.0 V
Assigned System	N/A
Pressure	0.0 psi

(Physical Component) EHSA Blue

Name	Value
Variable Domain	
Component Variables	
Assigned Surface	ELEV_1
Assigned Function	Pitch_P
Component ID	L_act1
Actuation Type	EHSA
Assigned System	H1, H3
Invoked Tool	ASSET
Tool Output	
Electric Power Demand	0.0
Maximum Hydraulic Flow Demand	2133.95 cm3/s
Weight	31.0 kg
Ground	
Hydraulic Flow Demand	213.9 cm3/s

(Physical Link) Hydraulic Power

Name	Value
Variable Domain	
Connector Variables	
Signal type	N/A
Voltage Type	N/A
Voltage	0.0 V
Assigned System	H1, H3
Pressure	3000.0 psi

Figure 3.36: Properties applied to physical components and physical links as a basis for inferential logic

Similarly, the assignment of actuators to control surfaces may be determined by iterating through all physical links. Figure 3.36 shows an example of an actuator and control surface combination. Here, the physical links represent the hydraulic power that is supplied to the actuator. Inspection of the properties contained within these links reveals additional details such as pressure and assigned hydraulic system identifiers.

Integration of extracted parameters into an MDAO workflow

For the example of the AGILE 4.0 project, the MDAO workflow specification is created collaboratively using the KADMOS tool [159] and the process platform KE-chain. In KE-chain, each tool owner specifies a so-called design competence. For each of these design competencies, the input and output parameters are initialized using separate CPACS files as an interface standard. Additional information, such as tool parameters, versioning, and ownership information, may also be provided at this stage. Following this stage, a merged CPACS baseline file is created, as shown in figure 3.37, by linking all the individual input and output files. This merged baseline carries the information between tools during workflow execution. The next steps involve modifying the workflow architecture, specifying the tool execution order, setting up design parameters, and specifying constraints, objectives, and state variables. A CMDOWS file is generated, which can then be used to execute the workflow in RCE along with the input baseline CPACS file.

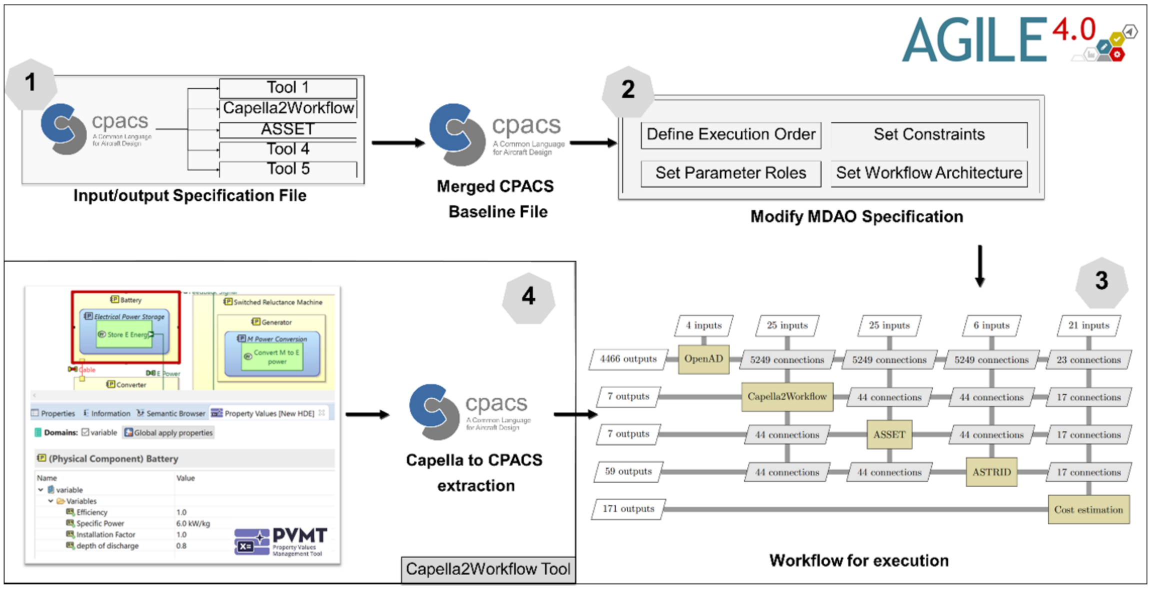


Figure 3.37: Process to integrate Capella with AGILE 4.0 workflow

The system architecture from figure 3.33 is used as input to the ASSET tool, which performs the actuator sizing and returns the overall actuator weight and power demand. This tool derives its input from the parameters specified in Capella. The functionality of parameter extraction is integrated within the AGILE workflow as the Capella2Workflow tool. This is performed by defining the input and output within the AGILE workbench. Capella2Workflow provides the parameter values directly to ASSET[10] (in development at Concordia University and described in [66, 160]), which further passes its output to the ASTRID tool (developed by the Politecnico di Torino [80]), which performs the sizing of all other systems. The parameters extracted from Capella are written directly to the output CPACS file of Capella2Workflow, which then serves as the input file for the ASSET tool. The workflow output is written to the baseline CPACS file. The relevant parameters are extracted and written back to the Capella data model schema, which is then available for inspection within the Capella workbench.

Discussion

Overall, the presented case study demonstrates how properties can be added to Capella model elements and how they can be extracted by parsing the data model schema. The

[10] ASSET: Aircraft System Sizing Estimation Tool

addition of properties is flexible, with the ability to define multiple property groups. In this study, property extraction is performed by manually parsing the Capella data model. However, the parsing approach is generic and can be performed manually or automatically with the appropriate tools. The challenge of automating this process is reduced by understanding and simplifying Capella's data schema structure.

An initial approach to integrating Capella, as an MBSE tool, with the system-level MDAO tools in the AGILE 4.0 workflow is also presented. The current approach is straightforward; it packages the extraction of Capella model properties as a tool that provides the properties as input to other tools in the workflow. The main limitation of this approach is that the system-level workflow, including the selected tools and execution order, needs to be predefined.

Chapter 4

Case Studies and Results

4.1 Overview

This chapter demonstrates the utility and features of the proposed framework through a series of case studies that cover each system architecting activity. The case studies help illustrate how safety is integrated into each system architecting activity and how the framework can be used to enable safety assessment in conceptual design. The case studies presented in this chapter are listed as follows:

1. Case studies 1 and 2: Rule-based design space filtering

2. Case studies 2 and 3: Early evaluation of system architectures for safety using quantitative safety assessment methods

3. Case studies 4 - 8: Architecting novel aircraft system architectures

4.2 Rule-based Design Space Filtering

This section presents two case studies demonstrating the filtering, using safety rules, of a large design space of candidate system architectures. The two filtering approaches discussed here are evaluative filtering and generative filtering. In evaluative filtering, the design space is first populated with candidate architectures using a system architecture definition tool and then evaluated using the safety rules. In generative filtering, the safety rules are used to build the system architectures from the bottom up by constraining the potential connections between system components. The case studies use a hydraulic landing gear braking system architecture where the design space is characterized by the choice of brake actuation technology, the allocation of brake actuators to each wheel, and the allocation of power supplies to each brake actuation unit.

4.2.1 Case Study 1 and 2

Each study comprises three activities: 1) Architecture Definition, 2) Representation and Rule Selection, and 3) Rule Evaluation. In the evaluative filtering approach, the Architecture Design and Optimization Reasoning Environment (ADORE) tool developed by Bussemaker et al. [51,161] is used to generate the system architecture design space. A wrapper for ADORE is then developed to convert each architecture in the design space into an equivalent graph-based architecture descriptor using generic elements. Finally, each graph is evaluated according to the applicable safety rules, and uncompliant architectures are filtered out of the design space.

Figure 4.1 illustrates the scope of the case study. A large design space for aircraft landing gear braking system architectures is generated using the ADORE tool; the resulting architectures are then filtered for feasibility based on safety heuristics. This determines whether an architecture in the design space conforms to a set of heuristics that represent implicit system safety.

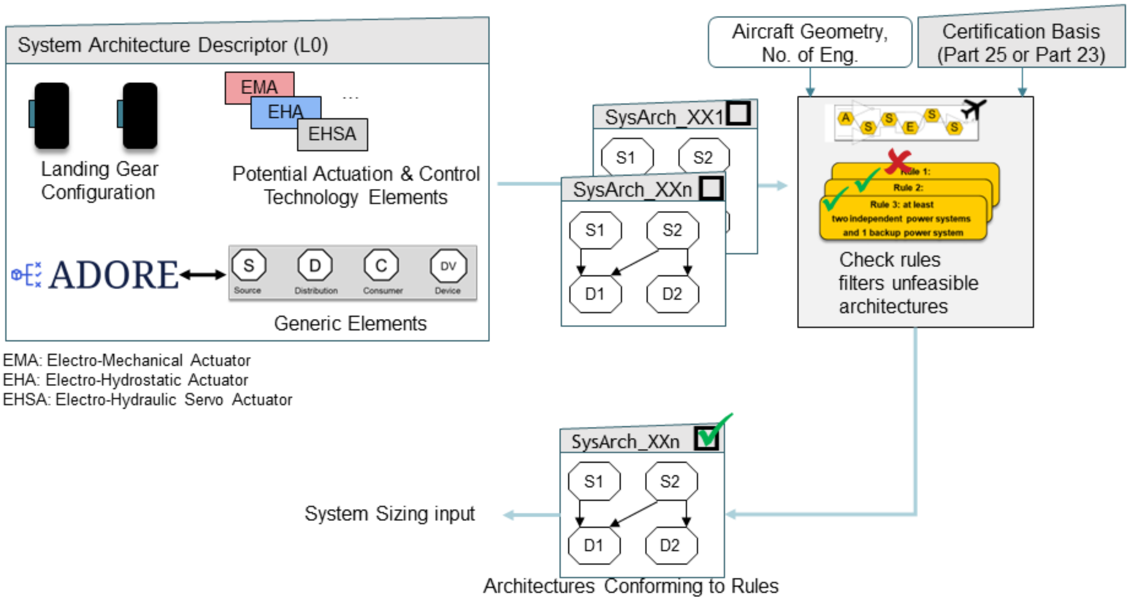

System Architecture Descriptor (L0)
Landing Gear Configuration
EMA
EHA
EHSA
...
Potential Actuation & Control Technology Elements
ADORE
S
Source
D
Distribution
C
Consumer
DV
Device
Generic Elements
EMA: Electro-Mechanical Actuator
EHA: Electro-Hydrostatic Actuator
EHSA: Electro-Hydraulic Servo Actuator
Aircraft Geometry, No. of Eng.
Certification Basis (Part 25 or Part 23)
SysArch_XX1
SysArch_XXn
S1
S2
D1
D2
A S S E S S
Rule 1:
Rule 2:
Rule 3: at least two independent power systems and 1 backup power system
Check rules filters unfeasible architectures
SysArch_XXn
System Sizing input
Architectures Conforming to Rules

Figure 4.1: Case study description: landing gear braking system with brake actuation technology choices evaluated for safety rules in the ASSESS tool, from [162]

In this case study, no feedback will be provided to ADORE for guiding the design space exploration, which is something that could be accomplished in other workflows by evaluating design objectives. A regional aircraft is used for these studies. The top-level requirements for these aircraft are at the border between Part 23 and Part 25 certification regulations and provide a favourable testbed for observing the impact of certification regulations on the overall aircraft design. This study focuses on the landing gear braking system, as it represents a system characterized by choices in actuation technology that result in the need for a power system, which, in turn, requires redundancies in selection and allocation. Furthermore, landing gear braking system requirements vary depending on the set of applicable certification rules (Part 25 or Part 23).

The landing gear configuration is assumed to be a tricycle type consisting of one nose gear and two wing-mounted main gears. Each main gear consists of two wheels, and each wheel is considered to have a single braking unit represented by a consumer element. The consumer element used to represent the wheel brakes is an abstraction of the components typically found in an aircraft wheel braking system. This simplified representation does not include sub-components such as discs (stators and rotors) of various types, valves, pistons and cylinders. Each braking unit taken as a whole differs in the type of power that it uses, and the components actuating the stator onto the rotor can be implemented as an EHSA, an EMA, or a local hydraulic generation-based braking unit. Therefore, each gear can feature combinations of braking units which use different actuation technologies. Exceptionally, in the case of fully electric actuation, each braking unit is assumed to comprise four individual EMAs.

4.2.1.1 Evaluative Filtering

The design space generation process in ADORE requires that an initial system architecture model be defined. This requires the identification of functions, components to fulfill those functions, and multiple options for function fulfillment. The implemented architecture design space model starts from the boundary function "provide wheel braking," which is specialized to the "provide wheel braking actuation" function. This function is implemented by either hydraulic or electric brake actuators, as shown in section 3 of figure 4.2. Both actuator types induce the function "provide deceleration," fulfilled by the wheels, and functions for providing their respective power types (i.e., hydraulic or electric). Providing power is fulfilled by the hydraulic and electric system, as shown in section 2 of figure 4.2. Power is generated by the engine or emergency power system, as shown in section 1 of the same figure. One actuator is installed per wheel, leading to 4 actuators in total. The distribution system components have been configured to present three instances, and the engine has two instances (representing two engines).

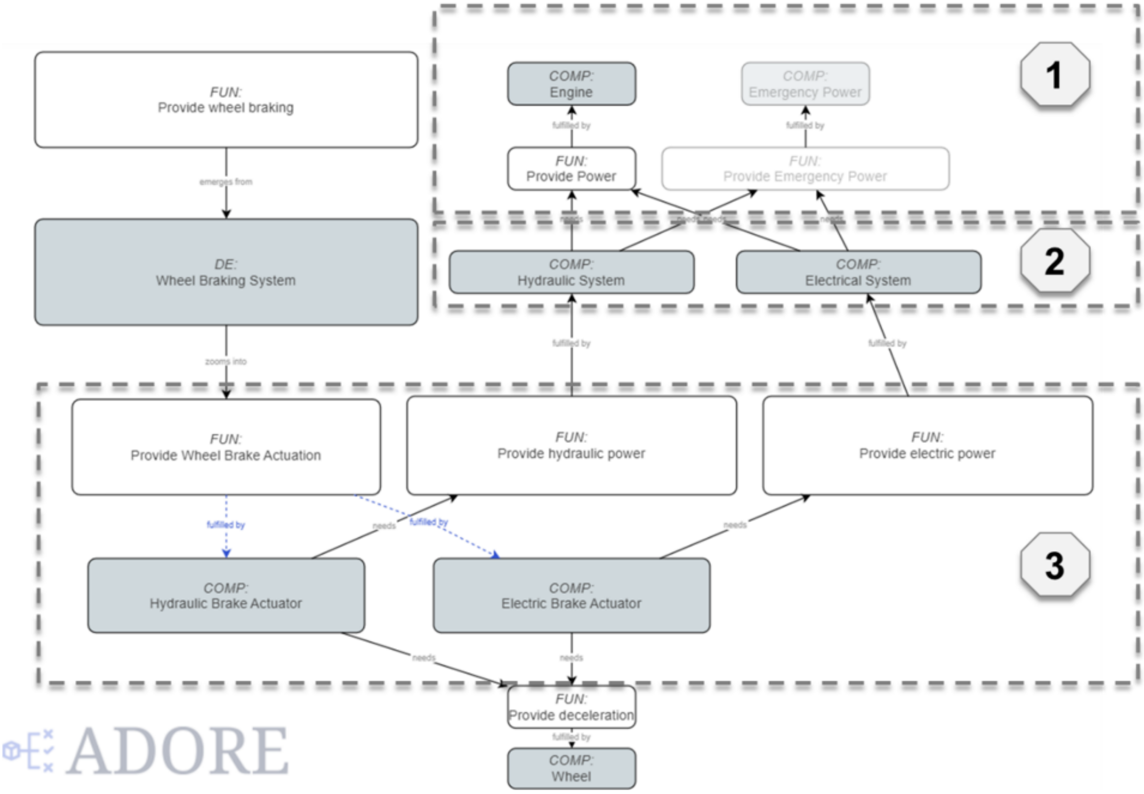


Figure 4.2: System architecture definition in ADORE, from [162]

Alongside the choice of the type of actuation to use, the most relevant choices are about how to connect the engines (power generation source) to the hydraulic or electrical system (power distribution) and then on to the actuators (consumers). These choices are implemented using ports.

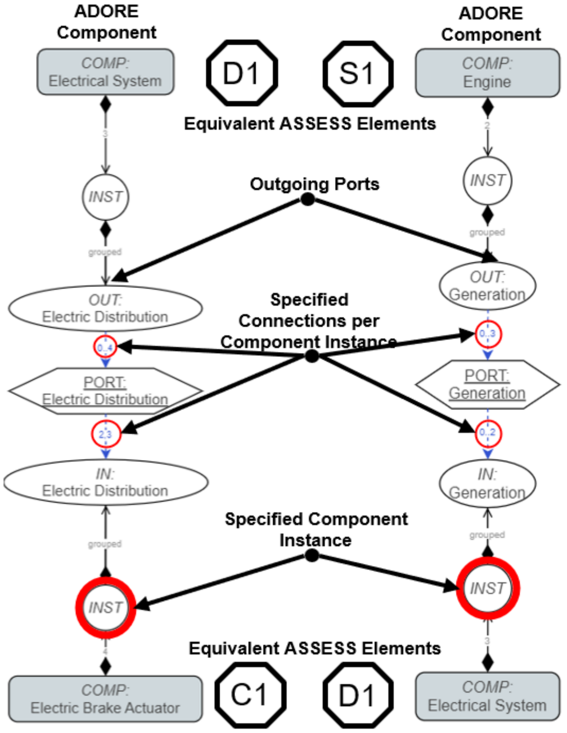


Figure 4.3: Port connections definition in ADORE and mapping to ASSESS elements, from [162]

Ports represent connection decisions from one or more outputs to one or more inputs. For example, from engines to hydraulic systems, the port represents a connection from 2 outputs (for two engines) to 3 inputs (for three hydraulic systems). Additionally, limits can be placed on the number of connections that each output or input port can establish or accept. In this case, each engine can be connected to none or any of the systems: each engine can establish between 0 and 3 connections, as shown in figure 4.3. Each hydraulic system needs to be connected to at least one source: their input ports accept between 0 and 2 connections from the engines; one input can also come from an emergency power generation system. In total, connections from engines to hydraulic systems can be established in 49 different patterns.

This section presents the results of the study and is structured as follows: First, an overview of the Design of Experiments (DOE) used to populate the design space using the ADORE model is presented. Second, the results of filtering the design space for hydraulic landing gear braking architectures are described, followed by results for electric landing gear braking. Finally, a specific case of building a design space using safety heuristics is shown.

Design of Experiments (DOE)

The Python library 'pymoo' [163] used within ADORE generates a DOE of different sample sizes using the Latin hypercube sampling algorithm. Using a DOE helps ensure good coverage of the design space while managing computational cost and time. Here, the ADORE model is provided as an input, and a design space is populated based on the points generated by the algorithm. Three discrete design variables are used to build the DOE. These are as follows:

1. The connection between the engine and distribution system, which accounts for 49

variations

2. The connection between the emergency (backup) generation system and distribution system, which accounts for 8 variations

3. The connection between the distribution system and the actuators, which accounts for 256 variations

Combining these numbers shows that a little over 100 000 different architectures can be generated. There is no design variable dependency, so this represents the complete combinatorial size of the design space. Table 4.1 presents the results of this study for the conventional landing gear braking case. Here, the DOE size is capped at 100, 5000, and 10000 architectures, respectively, for cases that include both Part 23 and Part 25 derived rules. A possibility of duplication exists for larger DOE sizes; this reduces the overall sample size. The rule-based filtering rendered 84% of the design space for the Part 25 case unfeasible. For Part 23, the stipulation of one main and one backup system per consumer results in a larger set of feasible architectures, with the rules filtering out 50% of the design space. Evaluating Rule 1 and Rule 2 on the connection between the distribution and consumer elements eliminates a third of the design space. The remainder is filtered out by checking if the connections between the source and distribution comply with Rule 1 (outlined in section 3.3.5).

Table 4.1: Rule-based design space filtering results for hydraulic (conventional) landing gear braking system architectures

DOE Size	Sample Size	Certification Basis	No. of Filtered Architectures	No. of Feasible Architectures	% of Design Space Deemed Feasible
100	100	Part 25	84	16	16.00
	100	Part 23	50	50	50.00
5000	4873	Part 25	4106	767	15.73
	4882	Part 23	2387	2495	51.10
10000	9473	Part 25	7966	1507	15.90
	9496	Part 23	4640	4856	51.13

Table 4.1 presents the results of this study for the conventional landing gear braking case. Here, the DOE size is capped at 100, 5000, and 10000 architectures, respectively, for cases that include both Part 23 and Part 25 derived rules. It is to be noted that a possibility of duplication exists for larger DOE sizes, which reduces the overall sample size. The rule-based filtering rendered 84% of the design space for the Part 25 case unfeasible. For Part 23, the stipulation of one main and one backup system per consumer results in a larger set of feasible architectures, with the rules filtering out 50% of the design space. Evaluating Rule 1 and Rule 2 on the connection between the distribution and consumer elements eliminates a third of the design space. The remainder is filtered out by checking if the connections between source and distribution comply with Rule 1.

Electrical Landing Gear Braking Case

Table 4.2 presents the results of this study for the electric landing gear braking case. The same DOE sizes are used as in the previous section for the Part 23 and Part 25 derived safety rule application. The rules filter out a maximum of 86% of the design space, which is more than for the hydraulic case, as the requirement of three main systems allocated per consumer element is more stringent than for the hydraulic case. Furthermore, ensuring the independence of electrical systems requires that an independent backup be supplied to at least one distribution element.

Table 4.2: Rule-based design space filtering results for electric landing gear braking system architecture

DOE Size	Sample Size	Certification Basis	No. of Filtered Architectures	No. of Feasible Architectures	% of Design Space Feasible
100	100 100	Part 25 Part 23	86 60	14 40	14.00 40.00
5000	4873 4889	Part 25 Part 23	4250 2950	623 1939	12.78 39.66
10000	9513 9489	Part 25 Part 23	8288 5678	1225 3811	12.87 40.16

4.2.1.2 Generative Filtering

The generative filtering or constrained design space analysis presents a bottom-up approach to safety-heuristic application. In this example, two source and hydraulic distribution elements are considered, and four hydraulic landing gear braking devices are used. Rule 1 constrains the connections between the distribution and consumer elements to reduce the combinatorial design space to one in which only connections between source and distribution elements need to be enumerated. Rule 1 ensures that each consumer element receives a connection from both distribution systems. To further simplify this scenario, an independent backup is assumed to be supplied to each consumer element as well.

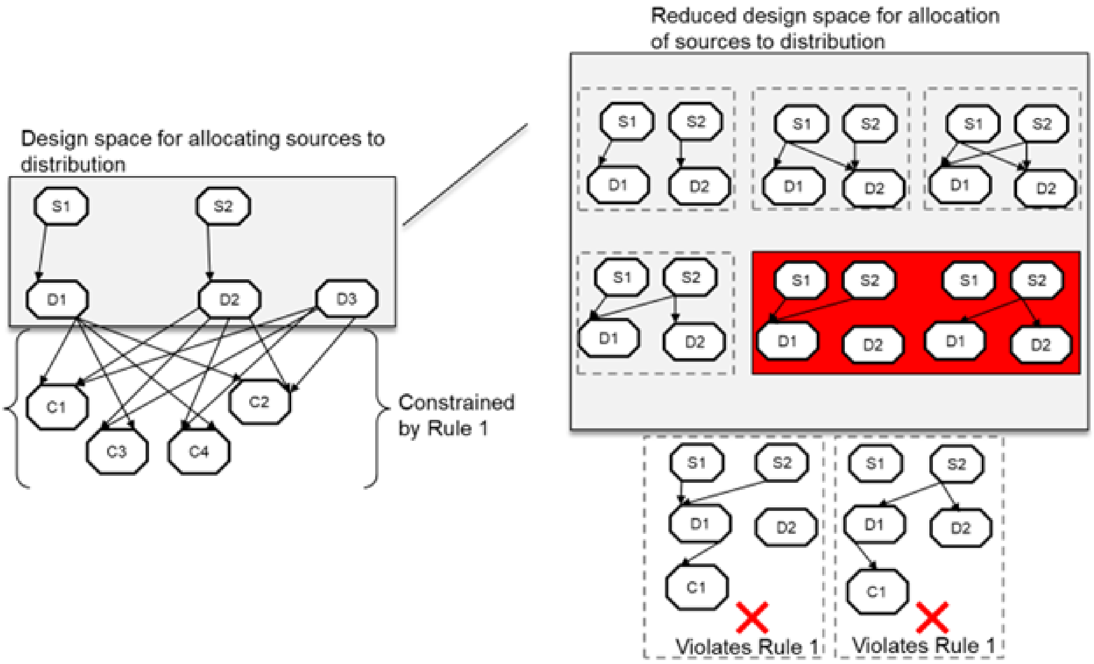
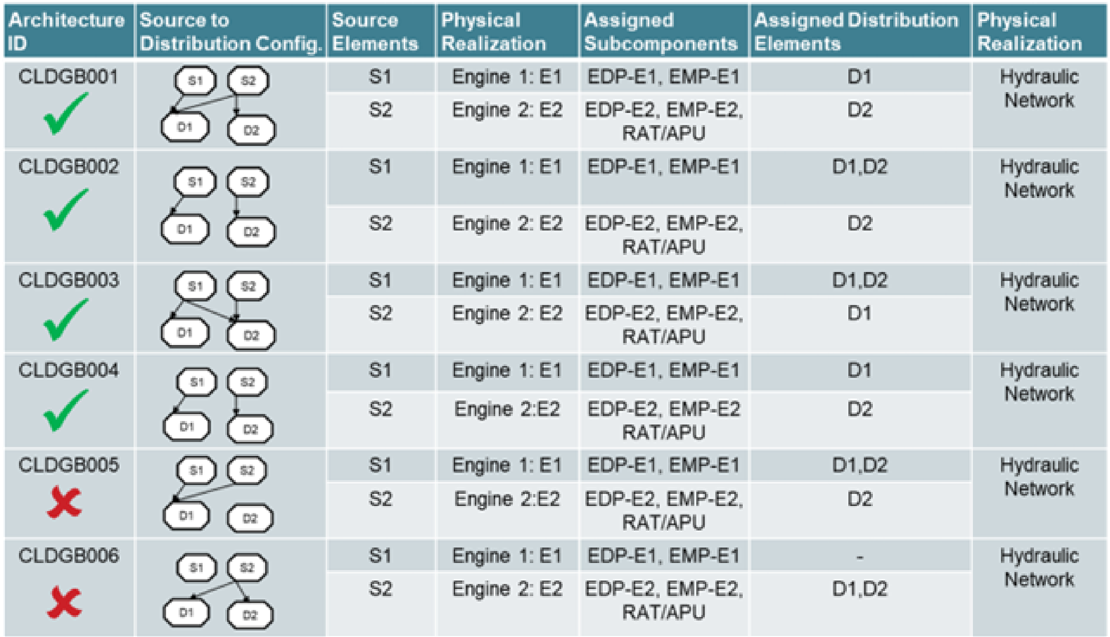

Reduced design space for allocation of sources to distribution
Design space for allocating sources to distribution
S1
S2
D1
D2
D3
C1
C2
C3
C4
Constrained by Rule 1
Violates Rule 1
Violates Rule 1

Figure 4.4: Generation of a constrained design space

Finally, the connections between source and distribution are enumerated, as shown in figure 4.4 for the hydraulic case. Once all possible connections are enumerated, Rule 1 is applied again to test if each source element is connected to an independent distribution element. This check reduces the design space to the four options shown in table 4.3 for the hydraulic braking system case. A similar result is observed for the electric landing gear braking case. Here, Rule 1 is applied under the special condition of having two distribution elements and an independent backup Source Element (SE) connected to at least one of the main distribution elements. Table 4.4 shows the reduced list of architectures conforming to the rules.

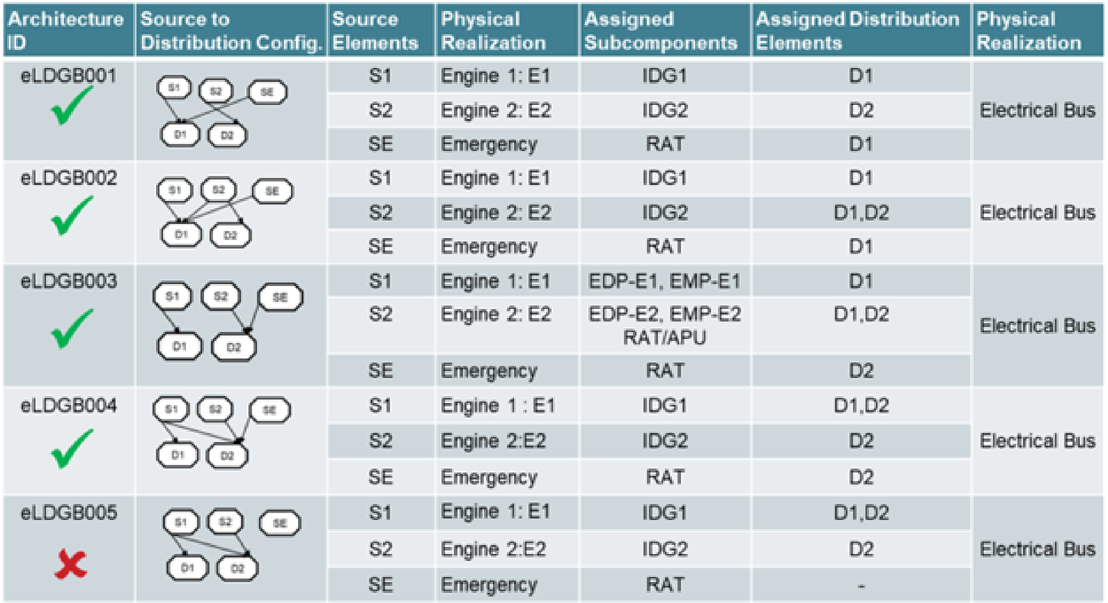
This reduced set of architectures can then be expanded by assigning physical components to each of the generic elements involved. This is shown in table 4.3 and table 4.4. Source elements are allocated individual engines, which are further allocated engine-driven pumps, while distribution elements are identified as electrical or hydraulic networks.

Table 4.3: Architectures generated based on safety rules for conventional hydraulic brake actuation



Architecture ID	Source to Distribution Config.	Source Elements	Physical Realization	Assigned Subcomponents	Assigned Distribution Elements	Physical Realization
CLDGB001 ✓		S1	Engine 1: E1	EDP-E1, EMP-E1	D1	Hydraulic Network
		S2	Engine 2: E2	EDP-E2, EMP-E2, RAT/APU	D2	
CLDGB002 ✓		S1	Engine 1: E1	EDP-E1, EMP-E1	D1,D2	Hydraulic Network
		S2	Engine 2: E2	EDP-E2, EMP-E2, RAT/APU	D2	
CLDGB003 ✓		S1	Engine 1: E1	EDP-E1, EMP-E1	D1,D2	Hydraulic Network
		S2	Engine 2: E2	EDP-E2, EMP-E2, RAT/APU	D1	
CLDGB004 ✓		S1	Engine 1: E1	EDP-E1, EMP-E1	D1	Hydraulic Network
		S2	Engine 2:E2	EDP-E2, EMP-E2 RAT/APU	D2	
CLDGB005 ✗		S1	Engine 1: E1	EDP-E1, EMP-E1	D1,D2	Hydraulic Network
		S2	Engine 2:E2	EDP-E2, EMP-E2, RAT/APU	D2	
CLDGB006 ✗		S1	Engine 1: E1	EDP-E1, EMP-E1	-	Hydraulic Network
		S2	Engine 2: E2	EDP-E2, EMP-E2, RAT/APU	D1,D2	

Table 4.4: Architectures generated based on safety rules for electrical brake actuation



Architecture ID	Source to Distribution Config.	Source Elements	Physical Realization	Assigned Subcomponents	Assigned Distribution Elements	Physical Realization
eLDGB001 ✓		S1	Engine 1: E1	IDG1	D1	Electrical Bus
		S2	Engine 2: E2	IDG2	D2	
		SE	Emergency	RAT	D1	
eLDGB002 ✓		S1	Engine 1: E1	IDG1	D1	Electrical Bus
		S2	Engine 2: E2	IDG2	D1,D2	
		SE	Emergency	RAT	D1	
eLDGB003 ✓		S1	Engine 1: E1	EDP-E1, EMP-E1	D1	Electrical Bus
		S2	Engine 2: E2	EDP-E2, EMP-E2 RAT/APU	D1,D2	
		SE	Emergency	RAT	D2	
eLDGB004 ✓		S1	Engine 1 : E1	IDG1	D1,D2	Electrical Bus
		S2	Engine 2:E2	IDG2	D2	
		SE	Emergency	RAT	D2	
eLDGB005 ✗		S1	Engine 1: E1	IDG1	D1,D2	Electrical Bus
		S2	Engine 2:E2	IDG2	D2	
		SE	Emergency	RAT	-	

Some advantages of this approach include the ability to quickly generate a reduced set of system architectures using the generic element representation. These can then be allocated physical components and evaluated within an MDAO workflow. Additionally, when a large number of systems are considered in conjunction, the overall complexity of the design space can be reduced by choosing to focus on a specific system for the complete enumeration of its design space. The constrained design space approach can then be used for each of the other systems.

Case Study 2: Validation using FTA

The architectures outlined in table 4.3 are evaluated using the Graph2FTA tool to determine the overall failure rate. Three typical scenarios of the loss of landing gear braking function are evaluated. These are an all-engine failure case, an all-actuator failure case, and the failure of an engine and the distribution associated with the opposite engine. The exponential distribution with a constant failure rate is used to model the reliability characteristics of different components with component failure rates provided in table B.1 of appendix B.1[1].

Table 4.5: Results of a quantitative analysis of selected architectures from table 4.3

Architecture	All Engine Failure (per FH)	All Actuator Failure (Per FH)	Single Engine and Opposite Distribution Failure (Per FH)
CLDGB001	1.42E-11	3.25E-12	5.47E-10
CLDGB002	1.42E-11	3.16E-12	5.47E-10
CLDGB003	1.42E-11	3.16E-12	5.47E-10
CLDGB004	1.42E-11	3.08E-12	5.47E-10

Table 4.5 shows that the architectures developed using the safety rules meet the quantitative requirement for a variety of critical failure conditions.

Specification and evaluation of the control architecture

After a safety check on the power architecture using the fault tree assessment has been performed, the control architecture can be specified. Since all four architectures have the same failure rate characteristics for the total loss of braking function, any of the four architectures can be selected for further development. In reality, aircraft-level metrics such as weight and fuel burn will need to be evaluated to quantify the impact of each architecture. This is not within the scope of the present study.

CLDGB001 is selected for further development of the control architecture. Initially, a single controller is represented using the "Ctrl" element and is connected to each actuator as shown in figure 4.5. The highlighted portion of figure 4.5 shows a single controller element being supplied through an electrical distribution system, which sources power from an emergency or backup power source as well as from an engine power source.

[1] Please refer to this appendix for a discussion on the collection of failure rate data.

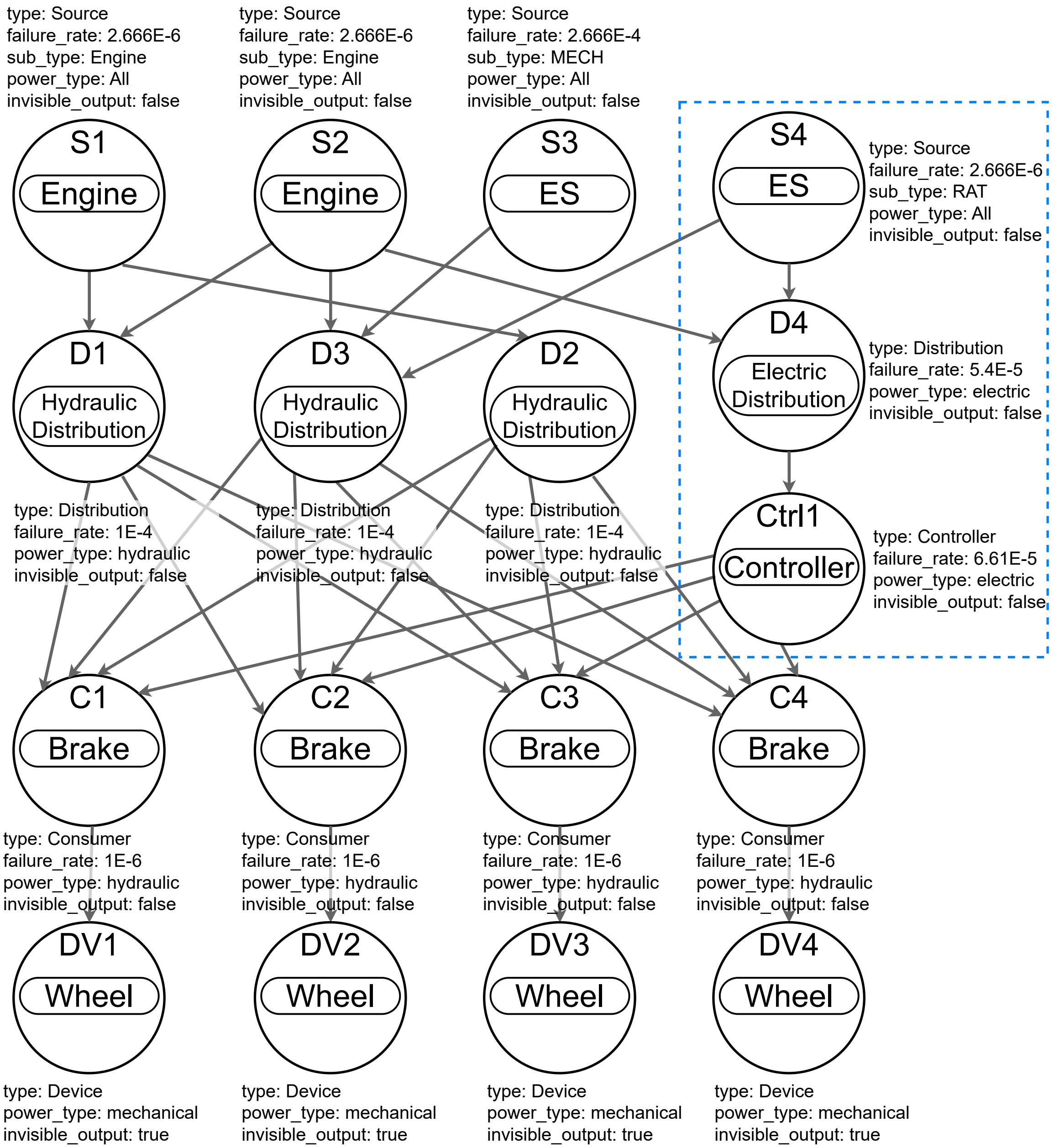
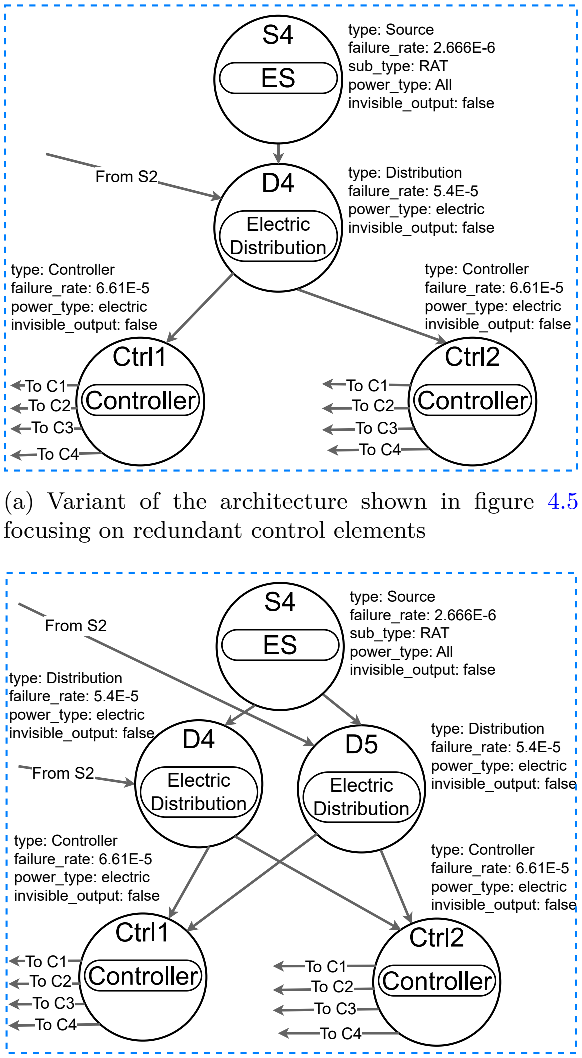


Figure 4.5: Representation of combined power and control view for evaluating the control architecture of an aircraft landing gear braking system

The architecture shown in figure 4.5 is modified by adding additional controller and electrical distribution elements to generate two variants. Figure 4.6(a) shows two controller elements supplied through a single electrical distribution system, whereas figure 4.6(b) shows two controllers supplied through two electrical distribution systems. These architecture variants and several others that feature three controllers and up to three distribution systems are evaluated using the graph2FTA tool to identify their safety characteristics using the number of failures per flight hour as a metric under different failure conditions. The different architectures are named using a schema that identifies the number of controllers and the number of electrical distribution systems in that particular architecture. An architecture named "CLGB001_1Ctl_1eDist" will feature one controller and one electrical distribution system; one named "CLGB001_3Ctl_2eDist" will have three controllers and two electrical distribution systems. The results of this evaluation are shown in table 4.6.



(a) Variant of the architecture shown in figure 4.5 focusing on redundant control elements

(b) Variant of the architecture shown in figure 4.5 focusing on redundant control and electrical distribution elements

Figure 4.6: Architecture variants created by adding redundant control and electrical distribution elements

All the architectures satisfactorily meet safety requirements for the "All source failure" and the "One source and opposite distribution system failure," i.e., the loss of braking function due to these failure scenarios is less than the order of 10^{-9}. This is due to the robust power generation and distribution architecture specified using the safety rules described above. However, the combined power and control views shown in figures 4.5, 4.6(a) and 4.6(b) indicate that the controller is now the key component as it contributes to a specific failure mode of the actuators, i.e., when the controller fails, the corresponding actuator is also regarded as having failed. Similarly, the controller is regarded as having failed when all distribution elements supplying it have been lost. Thus, the "All actuator failure" case is sensitive to redundancies in the controller and electrical power distribution elements.

Table 4.6: Failure rates for "Loss of all-wheel braking" function per FH

Architecture ID	Failure rate for "Loss of all wheel braking" function per FH		
	All source failure	One source distribution system failure	All brake actuator failure
CLGB001_1Ctl_1eDist	7.11E-12	1.44E-10	1.20E-04
CLGB001_1Ctl_2eDist	7.11E-12	1.44E-10	6.58E-05
CLGB001_2Ctl_1eDist	7.11E-12	1.44E-10	5.37E-05
CLGB001_2Ctl_2eDist	7.11E-12	1.44E-10	7.30E-09
CLGB001_3Ctl_1eDist	7.11E-12	1.44E-10	5.38E-05
CLGB001_3Ctl_2eDist	7.11E-12	1.44E-10	3.07E-09
CLGB001_3Ctl_3eDist	7.11E-12	1.44E-10	1.55E-10

Adding an additional electrical distribution improves the overall failure rate by an order of magnitude. However, for the "All actuator failure" condition, the loss of the controller and its internal failure rate plays the biggest role in the response of the architecture to this failure event. Adding an additional controller while keeping only a single distribution then makes the architecture sensitive to the reliability of the electrical distribution system. Additional redundancies in both electrical distribution systems and the controllers themselves are therefore required, such as in the two controller and two electrical distribution architecture in table 4.6. This architecture meets the required order of magnitude in the failure rate for the loss of function, but adding more controllers contributes to an overall improvement; the three controller and three distribution architecture exceeds the requirements by an additional order of magnitude.

4.2.1.3 Conclusions

This case study presents a practical approach to modelling and filtering large design spaces of candidate system architectures using safety heuristics. A key enabler is the link between the system architecture design space modelling tool ADORE and the rule-based safety assessment module of the ASSESS tool. The ASSESS tool features a set of safety heuristics derived from system architecture analysis, certification regulation, and industry best practices. These safety heuristics are formalized and used to filter a design space of conventional and electric landing gear braking system architectures. The formalized safety rules help reduce the design space significantly for both Part 23 and Part 25 certification cases. However, Part 23-based rules allow a much larger set of potentially feasible system architecture options. Additionally, an example of generative filtering was explored by constraining the design space using the aforementioned safety heuristics, resulting in a much smaller set of feasible architectures.

It is important to recognize the utility of the generative approach to building the system architecture using the safety rules. All the architectures developed using the rules possess desirable safety characteristics and also meet quantitative safety requirements in terms of redundancy in power distribution and allocation to the braking units. In the case of the control architecture, the safety rules are more conservative and result in an architecture with greater than the required redundancy in control elements. This shows that the rules can be used to build architectures that already have the requisite redundancy; an early quantitative assessment can help the system architect make informed changes to the architecture, including the reduction of redundant components in the aforementioned case of conservative specification.

Finally, it should be noted that emergent characteristics such as latent failures and other complex interactions are not considered in these early analyses and are not within the scope of the rules. Thus, this approach should only be used to do an initial check of required redundancy in order to specify an initial system architecture during the conceptual design stage. The benefits of this approach are that the architect can know if the specified architecture is sound from a safety perspective, if it has the requisite number of redundant components in the power and control architecture and if power and control are assigned to system components in a meaningful manner. This can help adjacent studies, such as weight and power estimation, which feed into the overall aircraft level performance estimation of an aircraft concept. In this way, the safety aspects are not neglected, thereby mitigating the risk of potential rework, reconfiguration and weight increase in later design stages.

4.2.2 Case Study 3: SOAPHiA Concept Aircraft

This study introduces a novel aircraft concept, shown in 4.7, that applies an incremental approach to aircraft electrification by combining systems electrification with propulsion electrification and the integration of a novel solar auxiliary power system. Electrified propulsion concepts for small regional or commuter aircraft have the drawback of incurring significant battery weight penalties due to the low energy density of current battery technologies. The SOAPHiA concept was introduced as a retrofitting option for the Dornier DO-228 aircraft to improve overall fuel burn reduction by using the incremental benefits of three types of electrification instead of relying on only propulsion electrification. A hybrid electric powertrain is used in conjunction with the electrification of the onboard system architecture, such as the environmental control system, and a third technology - an auxiliary solar power system (SPS) is introduced. The power output of the auxiliary SPS is used to reduce the power-off-take from the engine and supplements power supplied to the aircraft's systems. A detailed discussion of the SOAPHiA concept and an analysis of each type of electrification is provided in [164, 165].

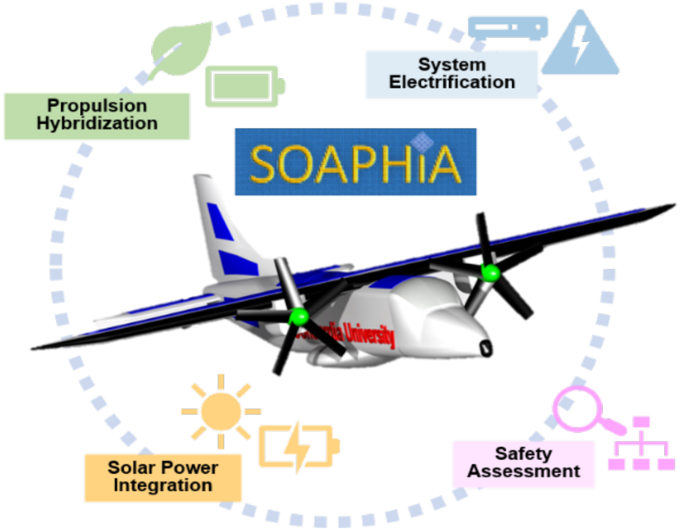


Figure 4.7: Overview of the Solar Auxiliary Power Hybrid Electric Aircraft (SOAPHiA), from [164]

As part of the investigation into systems electrification, both the electrification of the ECS and the FCS are considered. The following section outlines the effort to evaluate the safety characteristics of using electrical actuators for primary flight control on the SOAPHiA concept. The baseline Do-228 has mechanical flight controls, which are then replaced by fly-by-wire EHAs. This results in a reduction in fuel burn and can be beneficial in reducing overall CO2 emissions. However, electrical flight control actuation presents several challenges, one of them being the all-engine inoperative condition during which power must still be supplied to the flight control actuators.

This is achieved by configuring the architecture to have redundant and independent sources of electrical power. As such a design space can be explored with multiple electrical power sources in combination with a number of independent distribution systems. The roll control actuation system using the ailerons is developed using the generic element descriptors and evaluated to determine the overall system failure rate using the simple path approach.

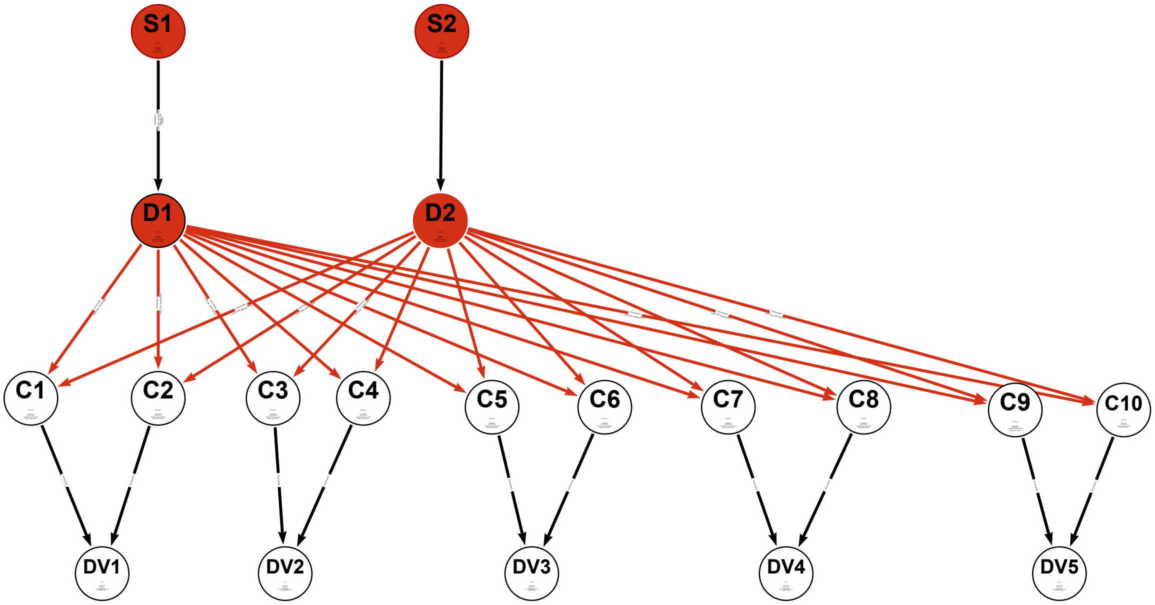


Figure 4.8: Impact of an all engine out condition on a roll control actuation and power supply system architecture with two main power sources

The Do-228 has two aileron surfaces and one generator associated with each engine. Each aileron surface is assigned a minimum of two EHAs. Three architectures are considered; the first uses the electrical power distribution system of the baseline Do-228 aircraft, which has two electrical sources and two distribution systems that provide electrical power to each EHA as shown in figure 4.8. The same figure also shows the elements that would be affected by the loss of both engine-based power sources, which in this case leads to the loss of the entire roll control function, resulting in the need for an additional or third source of electrical power.

The Do-228 can be configured with an optional APU, which can serve as the third independent source of power. As illustrated in figure 4.9, the installation of an APU represented by "S3" (in green) allows the aileron actuators to be supplied (through the blue connections) in case of an all-engine failure scenario.

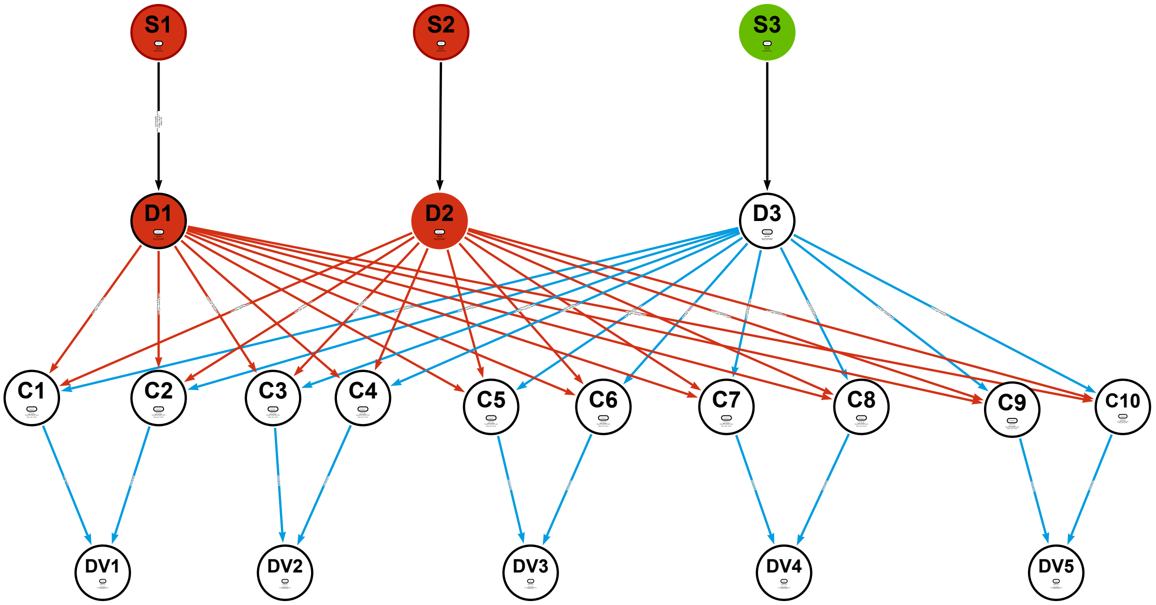


Figure 4.9: Roll control actuation and power supply architecture showing all engine out condition with additional source of electrical power

However, it is still possible to have two distribution systems and three electrical distributions. Therefore, the resulting network failure rates were computed for each option shown in Table 4.7.

Table 4.7: Options for electrical distributions and sources, network reliability, and % change in system weight

Option	Number of Electrical Distributions	Number of Electrical Sources	Network Reliability	% Change in System Weight
A	2	2	5.51×10^{-9} per FH	Baseline
B	3	2	5.50×10^{-9} per FH	14
C	2	3	4.10×10^{-13} per FH	13.16

The results indicate that in the specific case of this roll control actuation system architecture, adding redundancy in the electrical sources has a greater impact on the overall failure rate than when redundancy is added to the electrical distribution. In order to support the all-engine failure scenario, a minimum of three independent sources of electrical power are required. Option C meets the required safety requirements and is evaluated for the impact on system mass. This results in a systems weight increase of 13.16% over the baseline Do-228 with mechanical flight controls. The results suggest that an electrical actuation system is not beneficial towards overall systems weight reduction and subsequent fuel penalties for the Do-228. The aircraft would benefit from retaining the existing mechanical flight control actuation system while electrifying other power-consuming systems, such as the environmental control system and ice protection system.

Overall, this study demonstrates how the generic element descriptor can help the system architect build and gather safety insights into system architectures early in the design process.

4.3 Safety-focused Architecting of Systems Architecture for Novel Aircraft Configurations

4.3.1 Case Study 4: NASA Parallel Electric-Gas Architecture with Synergistic Utilization Scheme (PEGASUS) Concept

The NASA PEGASUS concept was developed based on the results of Antcliff et al., in [166], which outlined the potential performance benefits of electrification on regional aircraft with operating ranges of up to 400 nautical miles. The PEGASUS concept was presented in [167] and evaluated in several studies [168–171] with a recent update provided in [172] presenting the PEGASUS 2.0 concept. As shown in figure 4.10, the PEGASUS 2.0 concept is a 40-passenger aircraft similar to the ATR-42 but includes two inboard electric propulsor units and two wingtip hybrid-electric propulsor units in addition to an aft boundary layer ingestion propulsor. The wingtip propulsors are sized for cruise, and the inboard propulsors are used for take-off and landing while they conformally fold during cruise.

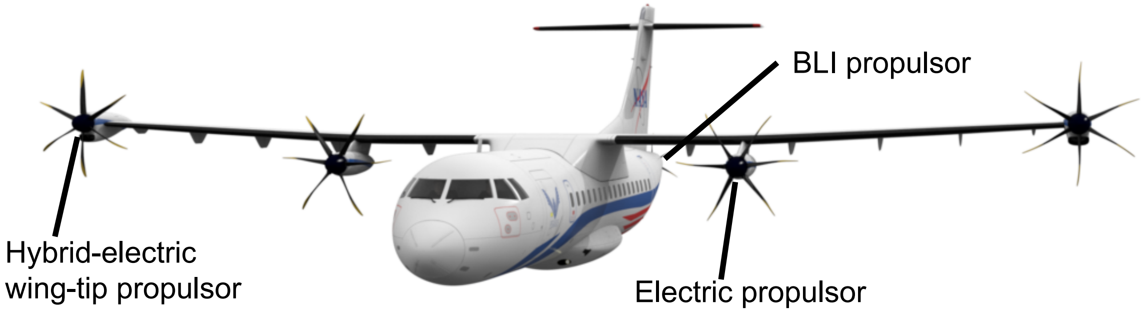


Figure 4.10: NASA Parallel Electric-Gas Architecture with Synergistic Utilization Scheme (PEGASUS) Concept, adapted from [173]

Blaesser et al. [172, pg .27] note that the PEGASUS 2.0 concept requires a larger vertical tail than the baseline aircraft in their study due to the potential of losing a wing-tip propulsion during take-off. This is driven by the asymmetric yawing moment that needs to be counteracted in this particular failure scenario. Blaesserr et al. [172, pg .28] highlight a potential area for exploration would be to use the wing tip propulsors primarily as control effectors, which could help reduce the required size of the inboard propulsors and also provide benefits in induced drag reduction.

The use of wingtip propulsors for yaw control requires the specification of a robust power system architecture. The wing-tip propulsors are assumed to provide the same function, "Provide yaw control," as the rudder surface. A critical loss of thrust (CLoT) of a wingtip propulsor can result in asymmetric yaw, which needs to be counteracted by the rudder surface to meet minimum control requirements and to preserve the integrity of the yaw control function. Thus the partial " loss of yaw control" due to CLoT will depend on the characteristics of the power architecture to the wing tip propulsors based on an assumption that the wing-tip propulsors are part of the propulsion power distribution system architecture which is powered from battery-based power sources. On the other hand, the total "loss of yaw control" will depend both on the aircraft power system architecture and the propulsion power system architecture. In this study the safety rules are applied to build the propulsion power system architecture and the partial " loss of yaw control" condition is evaluated.

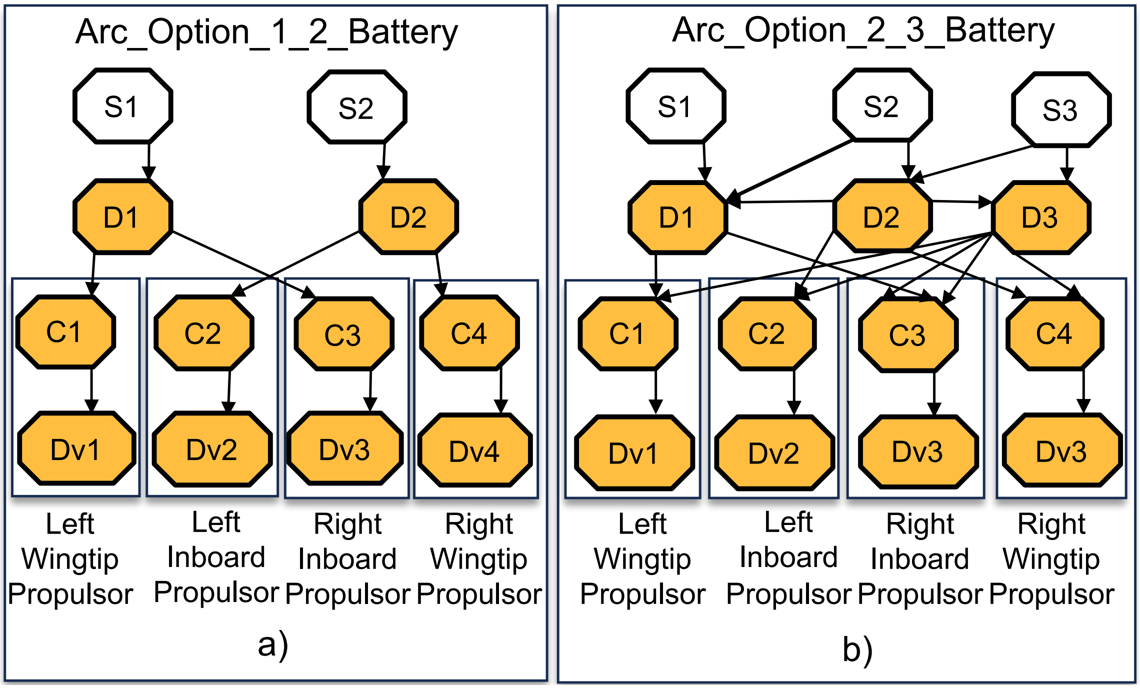


Figure 4.11: Notional propulsion electrical power distribution system architecture options for the NASA PEGASUS concept

Figure 4.11 shows two architecture options for the propulsion electrical power distribution architecture. Both options rely on batteries as the main source of power and provide power through distribution systems to the wingtip[2] and inboard propulsors. Option 1, with two batteries, is based on providing one power distribution lane for each battery and assigning each pair of wingtip propulsors and each pair of inboard propulsors to D1 and D2, respectively. Architecture option 2 is modelled based on the rules outlined for unconventional aircraft system architectures in section 3.3.3. The electric motors are represented as consumer elements in figure 4.11 that are assigned to device elements representing propellers. The architecture is built by ensuring that at least three non-repeating power paths from the source element, going through the distribution elements, terminate at each device element.

Both architectures are evaluated for functional failure condition, i.e., "Partial loss of yaw control due to CLoT," using the graph2AltaRica tool, which is part of the ASSESS L1-M1 tool suite. CLoT is divided into two failure events, CLoT-a being the loss of just the left wingtip propulsor [3] while CLoT-b is the loss of both the wing tip propulsor and the inboard propulsor on the left wing. In addition to these failure conditions, the loss of

[2] The gas turbine is not shown here

[3] The yaw control functionality provided by the wingtip propulsor is assumed to be lost regardless of the state of the gas turbine engine

all propulsors, the combination of the loss of battery power sources and the loss of the left wingtip propulsor, is also considered.

Typically, the criticality of the loss of each system function is established using an FHA. However, in this case, it is assumed that since the rudder will be sized to meet the requirements of CLoT, the partial loss of yaw control due to CLoT of a wing tip propulsor will not result in the aircraft being uncontrollable as long as the rudder is able to provide yaw control functionality. However, it may result in a significant increase in crew workload and a reduction in safety margins depending on the flight phase. Thus, erring on the conservative side, in this study, the partial loss of yaw control is classified as "Hazardous" and subject to a quantitative safety requirement of less than 10^{-7} failures per flight hour. The exposure time used in this study is two hours drawing from the endurance and typical operational characteristics of the ATR-42 aircraft on which PEGASUS 2.0 is based. Table 4.8 shows the failure rates for the partial function loss in question for each failure condition.

Table 4.8: Safety characteristics of each electrical propulsion architecture for different failure conditions

Architecture ID	Partial loss of yaw control			
	All Consumer Failure	Loss of left wingtip propulsor (CLoT - a)	Loss of left wingtip and inboard propulsor (CLoT - b)	Loss of all power sources and left wingtip propulsor
Arc. Option - 1 - 2 -Battery	3.36E-09	1.41E-04	3.98E-08	3.20E-09
Arc. Option-2-3-Battery	2.63E-13	9.95E-05	2.00E-08	7.68E-13
Arc. Option-2-3-Battery-Variant-1	2.57E-13	9.95E-05	2.00E-08	2.56E-13
Arc. Option-2-3-Battery-4-Distribution-Variant-2	2.57E-13	9.95E-05	2.00E-08	2.56E-13

Architecture option 1 with two batteries meets the requirements for CLoT-b but does not meet the safety requirement when evaluated for CLoT -a. The two battery power sources in architecture option 1 result in marginally acceptable values for the loss of all power sources coupled with the loss of a wing tip propeller. Architecture option 2, which is based on the safety rules, improves the overall failure rate across all cases except CLoTb. Both CLoT cases are limited by the conservative estimate of the electric motor failure rate selected for this study. Adding additional distribution elements and specifying added interconnections between the existing electrical distribution systems does little to improve the overall failure rate for CLoT a and CLoTb. Improving the failure rate of the wingtip drive by two orders of magnitude enables architecture option two to meet the safety requirement for all cases.

All architectures in this study feature more than one battery. This study does not take into account the sizing requirements for each battery in the architecture. It may be impractical to size each battery to the maximum power demand requirements, though it may still be possible to specify two batteries as primary batteries that together meet the power requirements of the motors. This approach can also be beneficial for system segregation requirements to protect against common cause failures such as propeller blade release. Furthermore, the third battery in architecture option 2 could be an emergency battery that is sized to provide backup power for an amount of time sufficient for the aircraft to be landed safely or for the pilots to restore primary power for electrical propulsion. The exact supply duration requirement for the emergency battery is subject to further analysis in conjunction with the aircraft operational characteristics, other certification requirements and feedback from the certifying authority.

At the conceptual stage, the architecture definition using safety rules, the architecture representation using generic elements and the architecture evaluation using quantitative safety rules demonstrated in this study help improve the confidence in the specified system architecture for such novel aircraft system architectures. The architecture specification developed here can now be used as input to system sizing and performance estimation tools to have an improved overview of the overall concept feasibility from a performance perspective.

4.3.2 Case Study 5: Novel Electric Motor Integration with Electrified Actuation (NOEMIE) Concept

The PEGASUS 2.0 concept features significant propulsion electrification through the use of hybrid-electric propulsors for cruise and fully electric propulsion for take-off and landing. This results in additional battery weight that needs to be carried, resulting in a larger MTOW than a similar regional aircraft baseline. Other challenges include the need for a larger vertical tail and additional sizing constraints of the inboard propulsor to meet criteria relating to the CLoT condition upon the loss of a wing tip propulsor. Blaesser et al. [172, p. 27] mention a direct correspondence between a reduction in wing tip drive thrust requirement and a reduction in vertical tail volume coefficient. Thus it could be advantageous to have a much smaller wing tip propulsor powered by an electrical power distribution system that would be supplied by smaller batteries. It may even be possible to reap the potential aerodynamic benefits of the wing tip drive while ensuring a manageable battery size which could further have a positive impact on integrating the batteries and the electrical system in the aircraft.

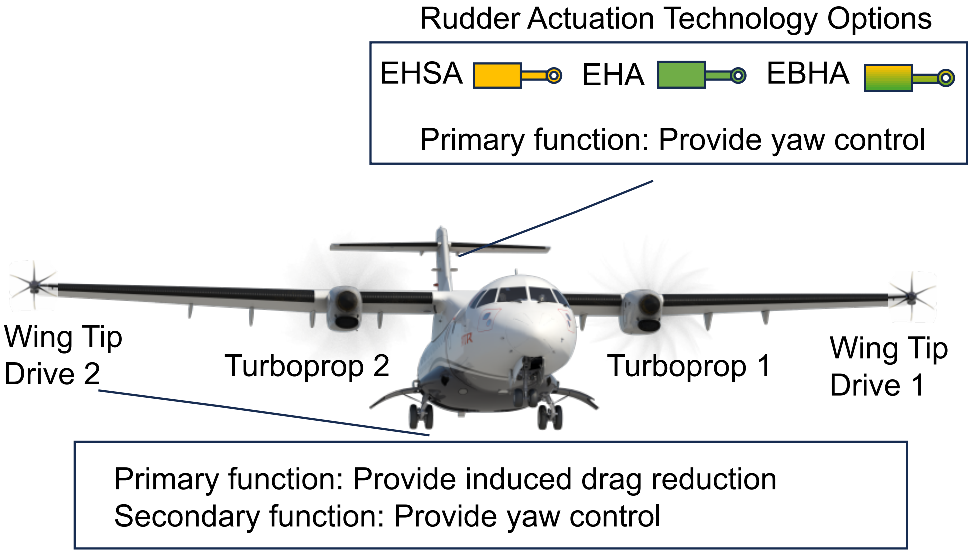

Rudder Actuation Technology Options
EHSA
EHA
EBHA
Primary function: Provide yaw control
Wing Tip
Drive 2
Turboprop 2
Turboprop 1
Wing Tip
Drive 1
Primary function: Provide induced drag reduction
Secondary function: Provide yaw control

Figure 4.12: Overview of the Novel Electric Motor Integration with Electrified Actuation (NOEMIE) aircraft concept

The Novel Electric Motor Integration with Electrified Actuation (NOEMIE) aircraft concept introduces electrified primary control surface actuation in addition to the integration of small wingtip motors, which are used through all flight phases to reduce the induced drag due to wing tip vortices. In addition to this function, they are also control effectors and are used for yaw control. The NOEMIE concept aircraft is based on the ATR-42 and retains the same engines turboprop engines as the baseline aircraft. The electric motors on the wing tip are sized for power levels significantly smaller than the turboprop engines. The thrust produced by these motors is small enough that the tail volume coefficient is within a similar range to that of the baseline ATR-42. This study does not provide the results of aircraft sizing but focuses on developing the system architecture outlined in section 4.3.1 in parallel with identifying the actuation system architecture for the aircraft rudder. The rudder and the wing tip propulsor together implement the "Provide yaw control" function.

Using the safety rules for yaw control actuation outlined in section 3.3.6, the rudder on the NOEMIE concept aircraft requires three actuators. Three actuation technologies, namely, EHSA, EBHA and EHA are available for use, giving rise to a possible 27 combinations of three of these actuators. The safety rules are also used to specify the number of power generation sources and distribution systems required based on the combination of actuators. To simplify the architectures that are generated, the number of electrical systems is set at three, while the number of required hydraulic systems is either two or three based on the combinations of actuator technology. The simple path approach outlined in section 3.6.2 is used to rapidly determine the overall system failure rate, which is shown in table 4.9 for the design space of rudder actuation system options considering an exposure time of two hours.

Table 4.9: Overview of system failure rates for a design space of rudder actuation options

Rudder Spot 1	Rudder Spot 2	Rudder Spot 3	Min. No. of Hydraulic Distribution Systems	Min. No. of Electrical Distribution Systems	Failure Rate per FH
EHSA	EHSA	EHSA	3	3	5.76823E-09
EHSA	EHSA	EBHA	3	3	1.22815E-11
EHSA	EHSA	EHA	2	3	2.45123E-10
EHSA	EBHA	EHSA	3	3	1.22815E-11
EHSA	EBHA	EBHA	2	3	5.33117E-11
EHSA	EBHA	EHA	2	3	6.31833E-10
EHSA	EHA	EHSA	2	3	2.58905E-10
EHSA	EHA	EBHA	2	3	3.40354E-10
EHSA	EHA	EHA	2	3	4.38921E-10
EBHA	EHSA	EHSA	3	3	4.07682E-11
EBHA	EHSA	EBHA	3	3	5.33117E-11
EBHA	EHSA	EHA	2	3	6.31833E-10
EBHA	EBHA	EHSA	2	3	5.33117E-11
EBHA	EBHA	EBHA	2	3	3.44465E-12
EBHA	EBHA	EHA	2	3	9.03792E-11
EBHA	EHA	EHSA	2	3	3.70346E-11
EBHA	EHA	EBHA	2	3	2.75679E-13
EBHA	EHA	EHA	2	3	6.27846E-11
EHA	EHSA	EHSA	2	3	2.58905E-10
EHA	EHSA	EBHA	2	3	3.70346E-11
EHA	EHSA	EHA	2	3	4.38921E-10
EHA	EBHA	EHSA	2	3	3.70346E-11
EHA	EBHA	EBHA	2	3	2.75679E-13
EHA	EBHA	EHA	2	3	4.87607E-11
EHA	EHA	EHSA	2	3	4.38921E-10
EHA	EHA	EBHA	2	3	6.27846E-11
EHA	EHA	EHA	2	3	4.38921E-10

All the analyzed architectures meet the safety objective of 10^{-9} failures per FH for the loss of yaw control prima facie. However, the simple path approach can easily be off by an order of magnitude even if a single common mode is not captured in the model, especially for unconventional architecture. Therefore the safety threshold is set as 10^{-10} failures per FH. This still leaves 26 possible architecture candidates; in this study, architecture 22 is selected for further study as it features one of each actuation technology. The generic element representation of architecture 22 is shown in figure 4.13.

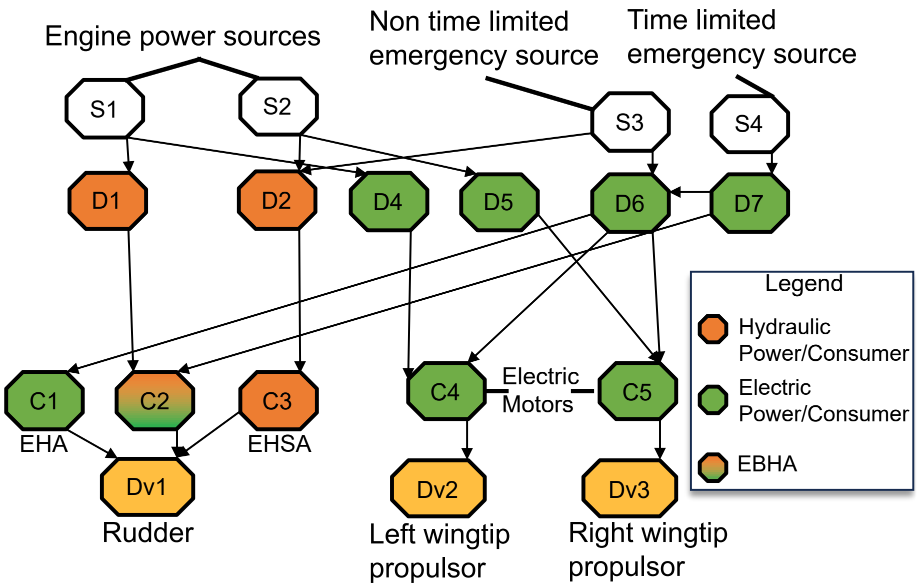


Figure 4.13: Representation of the power distribution architecture for rudder-based yaw control actuation integrated with wing tip propulsion-based yaw control for the NOEMIE concept aircraft

Here, the EHSA and EBHA actuators receive hydraulic power from two hydraulic systems that source hydraulic power from the aircraft's main engines and a non-time-limited emergency power source. The electrical power distribution architecture features four electrical power distribution elements, which together supply both the electrical actuators and the wingtip electric motors. The electrical sources comprise two engine-based power sources, a time-limited emergency source and a not time-limited emergency source. The emergency sources are routed through electrical distribution elements D6 and D7 and supply the electrical actuators and the electrical motors. The electric motors also receive power from D4 and D5 distribution elements that provide electrical power sourced from the engines.

The safety characteristics of this architecture need to be evaluated for a variety of potential failure conditions. The first three conditions are critical failures: loss of both engines, loss of an engine and the power distribution associated with the opposite engine, and loss of all actuators. In addition to these, asymmetric yaw due to CLoT of a wingtip drive needs to be considered. Since the wingtip propulsor and the rudder both provide yaw control, two additional cases of CLoT of a wingtip propulsion, along with the loss of all rudder actuators and the loss of two out of three rudder actuators, will also have to be considered. All the above-mentioned failure conditions, except the loss of a single wingtip propulsor due to CLoT, are expected to lead to the loss or significant degradation of the "Provide yaw control function", which is assumed to be classified as catastrophic[4]. The results of the safety-focused evaluation of the architecture are detailed in table 4.10.

[4] This classification is typically established through an FHA, which is not within the scope of this study.

Table 4.10: Safety characteristics of architecture option 22 from table 4.9 subject to different failure conditions

Architecture ID	Functional failure					
	Loss of yaw control					
Architecture option 22	All Engine Out	Single engine and opposite distribution	All actuator out	CLoT left wing tip drive	CLoT left wing tip drive and all actuator failure	CLoT left wing tip drive and two out of three actuator failure
Failure rate per FH	1.42E-11	5.33E-10	7.23E-21	9.95E-05	4.80E-18	7.12E-14

Architecture option 22 meets the safety requirement of $10^{-9}$for all critical failure conditions and falls short for the CLoT of the left wing tip drive, which results in a partial loss of yaw control but whose criticality is assumed to be major as in section 4.3.2. Sufficient redundancy is provided for both electrical and hydraulic actuators and the use of two power types for rudder actuation helps with exceeding the safety requirement for the single engine and opposite distribution case and the all actuator case. However, for the all actuator out case, care must be taken to identify all pertinent failure modes that could contribute to the failure of each type of actuator. In this study the all actuator failure is modelled as the loss of an actuator due to an internal failure or by a loss of power. Other failure modes may be dominant for each actuator, and therefore, this condition should be subject to more detailed investigation.

The loss of the left wing tip drive due to CLoT leading to asymmetric yaw is subject to the internal failure rate assumptions for the electric motor, which is set at a conservative 10^{-4}. Improved component reliability by two orders of magnitude will result in the architecture meeting the 10^{-7} requirement for the "Hazardous" failure classification. The CLoT of the left wing tip drive combined with the loss of all actuators is also subject to similar interpretation and further study as that of the all actuator out failure case. Finally, the CLOT of a wing tip drive, along with the loss of two out of three actuators, meets safety requirements. Here, the assumption is that the aircraft is still controllable and meets the minimum control requirements.

Overall, the system architecture studied here for the NOEMIE concept is promising from a safety perspective as it allows wing tip propulsor-based yaw control to be powered with the aircraft's electrical power system while simultaneously increasing the electrification of the yaw control actuation system. The architecture option 22 described in this study is suitable to be provided as input to a conceptual MDAO study to determine the aircraft-level feasibility of the NOEMIE concept. Though not within the scope of this study, this will be investigated in the future. The main takeaway from this study is that the three elements of safety-focused systems architecting, i.e, the rule-based architecture definition, design space quantitative safety evaluation and interactive architecture review and early safety analysis, can be successfully conducted in conceptual design for novel aircraft concepts and provide the system architect with sufficient insight to make safety-driven architecting decisions.

4.3.3 Case Study 6: Bombardier EcoJet

The EcoJet, Bombardier's research platform for the exploration of new technologies, features a Blended Wing Body (BWB) configuration [174]. The EcoJet platform described in this study is assumed to feature a dual rudder in an H-Tail configuration; this presents an interesting case for yaw control actuation system architecture trade studies from a safety perspective. Replacing The conventional single rudder with two rudder surfaces allows for a larger design space featuring variations in actuator technology, associated power system architecture, and the number of actuators allocated to each surface. A moderate design space of combinations exists, and several trade studies can be performed with safety as the primary criteria.

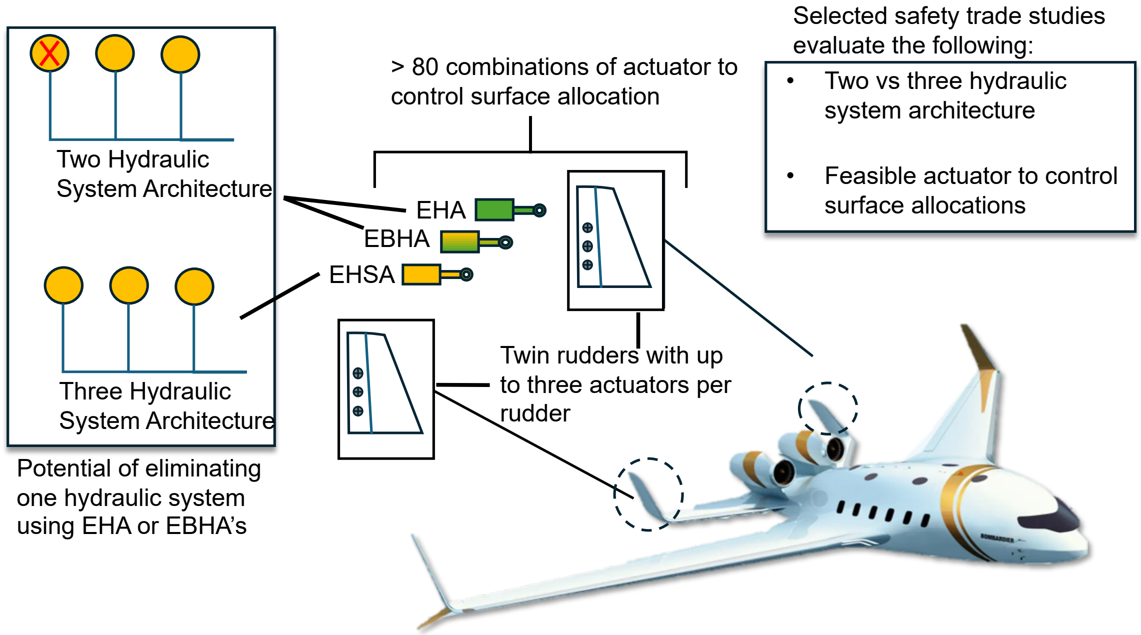

Two Hydraulic System Architecture
Three Hydraulic System Architecture
Potential of eliminating one hydraulic system using EHA or EBHA's
> 80 combinations of actuator to control surface allocation
EHA
EBHA
EHSA
Twin rudders with up to three actuators per rudder
Selected safety trade studies evaluate the following:
• Two vs three hydraulic system architecture
• Feasible actuator to control surface allocations

Figure 4.14: Overview of system architecting design space exploration for a notional yaw control actuation system of the Bombardier EcoJet

The following section presents two studies that demonstrate the applicability of the Graph to FTA and the path-based reliability assessment tool, respectively. These studies are illustrated in figure 4.14 and outlined as follows:

1. Comparison of two hydraulic system vs three hydraulic system architecture
2. Variation of actuation technology (EHSA, EHA & EBHA)
3. Specification of a notional fuel system architecture

Two vs Three Hydraulic Systems

The EcoJet features a twin vertical tail configuration that provides redundant surfaces for yaw control. Typical single rudder aircraft that use hydraulic actuation require three hydraulic systems for redundancy. However, now that additional redundancy is available in the number of surfaces themselves, there is potential to investigate using two hydraulic systems and thereby benefiting from potential weight savings.

The study is structured as follows: two base architectures are created, i.e., a two-hydraulic system architecture and a three-hydraulic system architecture. In both these architectures, two hydraulic actuators are allocated to each rudder surface, as shown in figure 4.15. The architectures are named according to their power distribution and allocation features. For example, an architecture named "Option_1_2_Hyd_2_con" implies an architecture with two hydraulic distribution systems and two consumers or actuators per surface. Each architecture is then evaluated for the "Complete loss of yaw control" function failure by considering the three critical cases of all engine failure, single-engine and opposite distribution system failure, and an all-actuator failure. Variations of the two baseline architectures that feature the additional consideration of potential power sources, such as an APU or Air Driven Pump (ADP), are also created and evaluated for the same failure conditions.

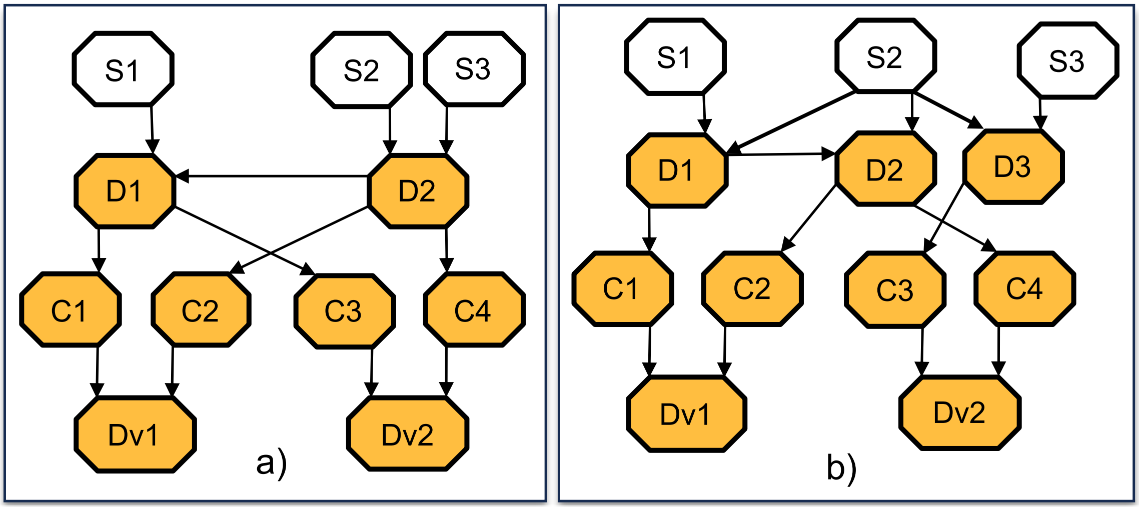


Figure 4.15: Overview of two baseline architectures featuring two and three hydraulic distribution systems respectively

Figure 4.15 a) shows a two hydraulic system variant where S1 and S2 are the hydraulic sources from the engine, and S3 is the emergency hydraulic power source from the RAT. The connection from D2 to D1 represents hydraulic power transfer using a component such as a PTU. Figure 4.15 b) shows a three hydraulic distribution system variant, with S1, S2, and S3 representing the same type of sources as in figure 4.15 a). The connection from D1 to D2 represents the PTU link, and the connection between S2 and both D1 and D3 can potentially represent two physical implementations. The first implementation is that S2 helps pressurize systems D1 and D2 by means of electrical power supplied to EMPs in each of the distribution systems. The second implementation could be that S2 has two engine-driven pumps, with each pump linked to one hydraulic system (D1 and D3). In both figures, Dv1 and Dv2 represent the rudder surfaces.

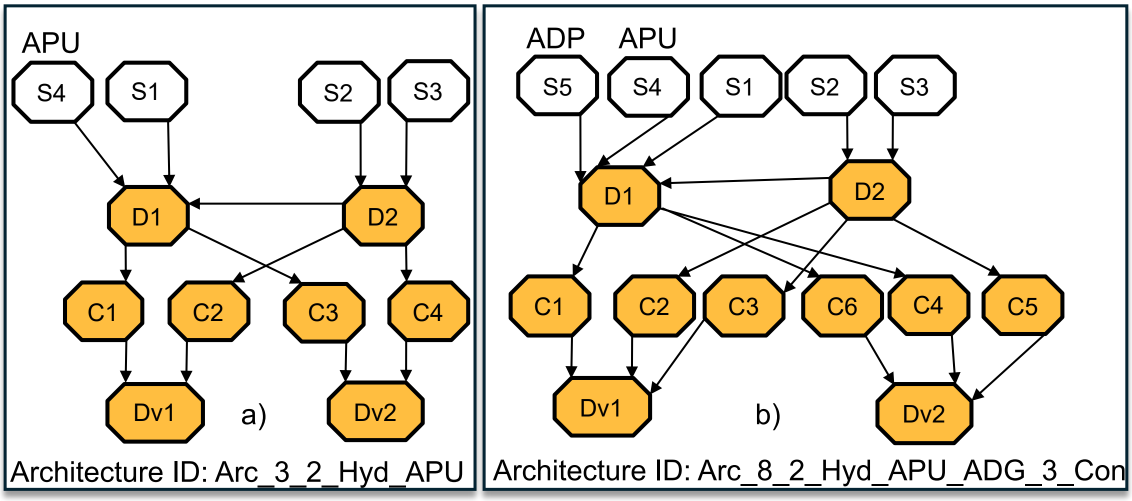

APU
S4
S1
S2
S3
D1
D2
C1
C2
C3
C4
Dv1
a)
Dv2
Architecture ID: Arc_3_2_Hyd_APU
ADP
APU
S5
S4
S1
S2
S3
D1
D2
C1
C2
C3
C6
C4
C5
Dv1
b)
Dv2
Architecture ID: Arc_8_2_Hyd_APU_ADG_3_Con

Figure 4.16: Overview of two variant architectures featuring the addition of an APU and ADG, respectively

Two further variants of the baseline architectures are shown in figure 4.16. In figure 4.15 b), S5 represents the hydraulic power from an ADG; In figure 4.15 a), S4 represents an APU-based hydraulic power source. The architecture shown in figure 4.15 b) also has three hydraulic actuators allocated to each surface. Table 4.11 shows the list of architectures that are evaluated with the corresponding failure rates for each failure condition.

Table 4.11: Summary of safety characteristics of different yaw control actuation architectures

Architecture ID	Failure rate for "Loss of yaw control" per FH		
	All source failure	One source distribution system Failure	All actuator failure
Arc_1_2_Hyd_Dist	3.55E-11	1.33E-09	5.13E-08
Arc_2_3_Hyd_Dist_b	3.54E-11	1.33E-09	3.10E-11
Arc_3_2_Hyd_APU	3.55E-11	1.33E-09	5.00E-08
Arc_4_2_Hyd_Multi_Pump_1	3.55E-11	1.33E-09	5.00E-08
Arc_5_2_Hyd_APU_Con_Rel	3.55E-11	1.33E-09	5.00E-08
Arc_6_2_Hyd_APU_3_Con	3.55E-11	1.33E-09	5.00E-08
Arc_7_2_Hyd_APU_ADG _3_Con	4.73E-14	1.33E-09	5.00E-08
Arc_8_2.5_Hyd_APU_3_Con	3.55E-11	1.33E-09	2.50E-11
Arc_9_2.5_Hyd_APU_2_Con	3.55E-11	1.33E-09	2.55E-11
Arc_10_2.5_Hyd_APU_2_Con	7.11E-11	1.33E-09	2.62E-11

The two hydraulic system architecture is limited by the all actuator failure case, where it does not meet the minimum safety objective. The three hydraulic system architecture

"Arc_2_3_Hyd_Dist_b" meets the safety objectives for all the failure conditions as expected. Attempts at addressing the drawbacks of the two hydraulic system architecture by creating variants that feature the use of APU hydraulic power source, the use of multiple primary hydraulic pumps, allocating three actuators to each device, increasing the assumed failure rate of the actuators, and using an air driven pump fail to make any impact on meeting the safety requirements specifically for the all actuator failure case.

Architectures "Arc_8" to "Arc_10" feature what is termed here as a 2.5 hydraulic system architecture. This is similar to the hydraulic system and rudder actuation implementation on the Boeing 737 aircraft, where systems A and B power a dual rudder actuator while a dedicated system, C, exists to supply power only to a backup rudder actuator.

"Arc_8", "Arc_9", and "Arc_10" are three variants of the 2.5 hydraulic system architecture. "Arc_8" features two main distributions and a backup distribution solely for specific rudder actuators while maintaining three actuators per control surface. "Arc_9" has the same 2.5 hydraulic distribution system architecture as "Arc_8" but differs by having two actuators per surface. "Arc_10" is similar to "Arc_9" but has additional connections between the distribution elements.

The results of the fault tree assessment shown in table 4.11 indicate that a two-hydraulic system architecture is sensitive to the all actuator failure case. This failure condition is directly addressed by adding redundancy in the hydraulic distribution system, by adding a third distribution. Further, architectural choices such as adding additional sources of hydraulic power, enabling power transfer between the different systems, or increasing the number of redundant actuators per surface have no effect on the safety characteristics of the two hydraulic system architecture for the specific case of an all actuator failure.

However, there is potential for implementing two main hydraulic systems and a backup system that is sized specifically to supply hydraulic power to one rudder actuator. "Arc_8", "Arc_9", and "Arc_10" show favourable improvements in the overall failure rate across the board and specifically in exceeding the quantitative safety requirements for the all actuator failure rate by two orders of magnitude. Such an architecture is not entirely novel as it is already certified aboard the Boeing 737. Its application to the EcoJet aircraft could be beneficial, provided that additional studies on weight, power consumption and aircraft-level impact also yield favourable results.

Variation of actuation technology

This study aims to explore the design space of possible combinations of actuation technologies for the actuators assigned to the two rudder surfaces on the EcoJet. The methodology used in this study is similar to that applied in section 4.3.2. First, a design space of possible actuator assignments (and corresponding actuation technology) to each rudder surface is generated. Two actuators are considered per surface; with three actuation technology options (EHSA, EBHA, and EHSA), this leads to a total of 81 candidate architectures. Second, these candidate architectures are then evaluated using the simple path approach of ASSESS L1-M1 and are filtered according to the failure rates obtained for each architecture. Third, a promising architecture from the initial reliability assessment is selected and further evaluated using the Graph2FTA method. Typically, a system sizing tool would also be incorporated to determine the aircraft-level impact of each architecture; this would be another criterion for eliminating candidate architectures and selecting a top-performing system architecture from both safety and aircraft-level weight and performance impact standpoints. However, since the objective of this case study is to demonstrate the utility of incorporating safety studies in early design space exploration by using the methodology developed in chapter 3, the evaluation of weight and other aircraft-level metrics are not performed and are considered outside the scope of this study.

Table 4.12 presents the architecture design space along with the results of the path-based assessment of the overall failure rate. A two-hydraulic and two-electric system architecture was adopted for all architectures except Architecture 1. Here, a two-hydraulic system architecture was considered and was found to not have the required redundancy to meet the failure rate requirement of less than 10^{-9} failures per flight hour. All 2H-2E architectures also have a third backup or emergency electrical system connected to a non-time-limited electrical source such as a RAT. A total of 44 architectures were found to have the required redundancy to meet the failure rate. In order to account for potential inaccuracy in the computation of the failure rate and the possibility of this early analysis missing critical failure modes, around 10 architectures that exhibited failure rates in the order of magnitude of 10^{-9} per FH were considered to be marginal cases and were not included in the final list of selected architectures.

Table 4.12: Overview of the design space of rudder actuation architectures featuring different actuation technologies with corresponding failure rates evaluated using the simple path approach

Architecture ID	Rudder 1 Spot 1	Rudder 1 Spot 2	Rudder 2 Spot 1	Rudder 2 Spot 2	Overall Failure Rate
1	EHSA	EHSA	EHSA	EHSA	2.26085E-05
2	EHSA	EHSA	EHSA	EHA	8.48522E-07
3	EHSA	EHSA	EHSA	EBHA	3.00136E-07
4	EHSA	EHSA	EHA	EHSA	6.93457E-07
5	EHSA	EHSA	EHA	EHA	3.12528E-08
6	EHSA	EHSA	EHA	EBHA	1.11072E-08
7	EHSA	EHSA	EBHA	EHSA	2.95664E-07
8	EHSA	EHSA	EBHA	EHA	1.33706E-08
9	EHSA	EHSA	EBHA	EBHA	4.75239E-09
10	EHSA	EHA	EHSA	EHSA	8.48522E-07
11	EHSA	EHA	EHSA	EHA	3.81905E-08
12	EHSA	EHA	EHSA	EBHA	1.35723E-08
13	EHSA	EHA	EHA	EHSA	3.12528E-08
14	EHSA	EHA	EHA	EHA	1.41654E-09
15	EHSA	EHA	EHA	EBHA	5.03523E-10
16	EHSA	EHA	EBHA	EHSA	1.33706E-08
17	EHSA	EHA	EBHA	EHA	6.06117E-10
18	EHSA	EHA	EBHA	EBHA	2.15451E-10
19	EHSA	EBHA	EHSA	EHSA	3.00136E-07
20	EHSA	EBHA	EHSA	EHA	1.35723E-08

Continued on next page

Table 4.12: Overview of the design space of rudder actuation architectures featuring different actuation technologies with corresponding failure rates evaluated using the simple path approach (Continued)

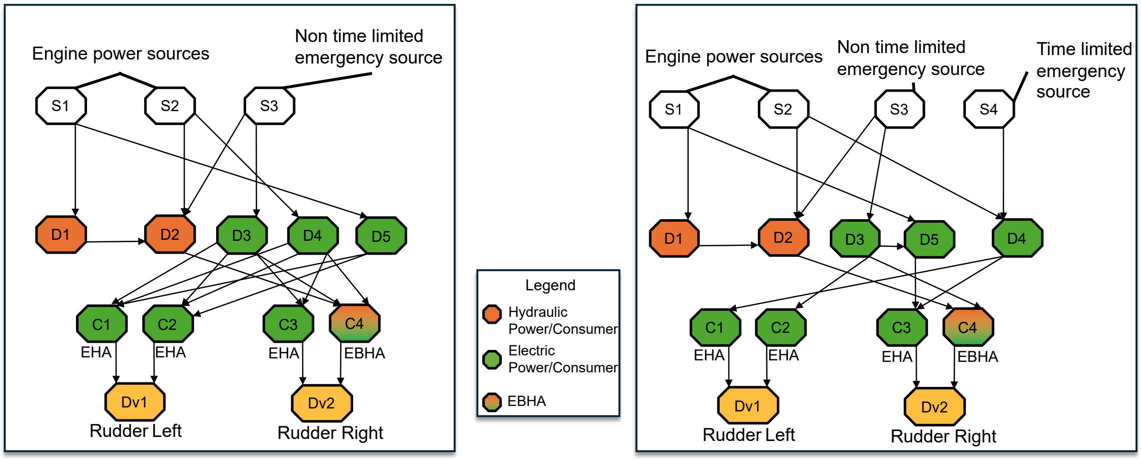
21	EHSA	EBHA	EHSA	EBHA	4.82407E-09
22	EHSA	EBHA	EHA	EHSA	1.11072E-08
23	EHSA	EBHA	EHA	EHA	5.03523E-10
24	EHSA	EBHA	EHA	EBHA	1.78983E-10
25	EHSA	EBHA	EBHA	EHSA	4.75239E-09
26	EHSA	EBHA	EBHA	EHA	2.15451E-10
27	EHSA	EBHA	EBHA	EBHA	7.65849E-11
28	EHA	EHSA	EHSA	EHSA	6.93457E-07
29	EHA	EHSA	EHSA	EHA	3.12528E-08
30	EHA	EHSA	EHSA	EBHA	1.11072E-08
31	EHA	EHSA	EHA	EHSA	2.55757E-08
32	EHA	EHSA	EHA	EHA	1.15928E-09
33	EHA	EHSA	EHA	EBHA	4.12078E-10
34	EHA	EHSA	EBHA	EHSA	1.09421E-08
35	EHA	EHSA	EBHA	EHA	4.96041E-10
36	EHA	EHSA	EBHA	EBHA	1.76324E-10
37	EHA	EHA	EHSA	EHSA	3.12528E-08
38	EHA	EHA	EHSA	EHA	1.41654E-09
39	EHA	EHA	EHSA	EBHA	5.03523E-10
40	EHA	EHA	EHA	EHSA	1.15928E-09
41	EHA	EHA	EHA	EHA	5.25581E-11
42	EHA	EHA	EHA	EBHA	1.86824E-11
43	EHA	EHA	EBHA	EHSA	4.96041E-10
44	EHA	EHA	EBHA	EHA	2.2489E-11
45	EHA	EHA	EBHA	EBHA	7.99402E-12
46	EHA	EBHA	EHSA	EHSA	1.11072E-08
47	EHA	EBHA	EHSA	EHA	5.03523E-10
48	EHA	EBHA	EHSA	EBHA	1.78983E-10
49	EHA	EBHA	EHA	EHSA	4.12078E-10
50	EHA	EBHA	EHA	EHA	1.86824E-11
51	EHA	EBHA	EHA	EBHA	6.64092E-12
52	EHA	EBHA	EBHA	EHSA	1.76324E-10

Continued on next page

Table 4.12: Overview of the design space of rudder actuation architectures featuring different actuation technologies with corresponding failure rates evaluated using the simple path approach (Continued)

53	EHA	EBHA	EBHA	EHA	7.99402E-12
54	EHA	EBHA	EBHA	EBHA	2.84158E-12
55	EBHA	EHSA	EHSA	EHSA	2.95664E-07
56	EBHA	EHSA	EHSA	EHA	1.33706E-08
57	EBHA	EHSA	EHSA	EBHA	4.75239E-09
58	EBHA	EHSA	EHA	EHSA	1.09421E-08
59	EBHA	EHSA	EHA	EHA	4.96041E-10
60	EBHA	EHSA	EHA	EBHA	1.76324E-10
61	EBHA	EHSA	EBHA	EHSA	4.68176E-09
62	EBHA	EHSA	EBHA	EHA	2.1225E-10
63	EBHA	EHSA	EBHA	EBHA	7.54469E-11
64	EBHA	EHA	EHSA	EHSA	1.33706E-08
65	EBHA	EHA	EHSA	EHA	6.06117E-10
66	EBHA	EHA	EHSA	EBHA	2.15451E-10
67	EBHA	EHA	EHA	EHSA	4.96041E-10
68	EBHA	EHA	EHA	EHA	2.2489E-11
69	EBHA	EHA	EHA	EBHA	7.99402E-12
70	EBHA	EHA	EBHA	EHSA	2.1225E-10
71	EBHA	EHA	EBHA	EHA	9.62281E-12
72	EBHA	EHA	EBHA	EBHA	3.42055E-12
73	EBHA	EBHA	EHSA	EHSA	4.75239E-09
74	EBHA	EBHA	EHSA	EHA	2.15451E-10
75	EBHA	EBHA	EHSA	EBHA	7.65849E-11
76	EBHA	EBHA	EHA	EHSA	1.76324E-10
77	EBHA	EBHA	EHA	EHA	7.99402E-12
78	EBHA	EBHA	EHA	EBHA	2.84158E-12
79	EBHA	EBHA	EBHA	EHSA	7.54469E-11
80	EBHA	EBHA	EBHA	EHA	3.42055E-12
81	EBHA	EBHA	EBHA	EBHA	1.21588E-12

Architecture 42 is selected for a more detailed assessment using the Graph2AltaRica tool. The basis of this selection within the scope of the present study is the will of the architect; in proper practice, other analyses such as weight and performance would also be criteria for selecting a promising candidate architecture.



a) Architecture 42

b) Architecture 42 modified to accommodate an additional emergency power source

Figure 4.17: Notional yaw control actuation architectures for the EcoJet

Figure 4.17 a) shows architecture 42 using the generic element descriptor. It exhibits a large number of connections from electrical distribution systems to the consumer elements (the electrically powered actuators). Some of these connections may be difficult to integrate based on component-level constraints. However, it is possible to simplify these connections and still maintain equivalent safety characteristics. In figure 4.17 b), a variant of architecture 42 is shown where at least one consumer on each surface receives power through an electrical distribution directly linked to a non-time limited power source. Furthermore, an additional time-limited source is introduced as an emergency source and is linked through a dedicated electrical bus to at least one electrical actuator on each surface. Table 4.13 shows that both architectures demonstrate equivalent characteristics when evaluated for different failure events.

Table 4.13: Overview of failure rate for the "Loss of yaw control" function for a pair of notional yaw control actuation architectures for the EcoJet

Architecture ID	Failure rate for "Loss of yaw control" per FH		
	All source failure	One Source Distribution system failure	All actuator failure
Architecture 42	3.54E-11	1.33E-09	4.71E-16
Architecture 42 modified	3.54E-11	2.65E-09	5.45E-18

Overall, the generic element-based architecture representation lends itself well to the development of novel system architecture design spaces and can be applied to developing system architectures for unconventional aircraft. Furthermore, the quantitative safety assessment approaches tailored to the information contained in the descriptor can work both to help filter a large design space and to foster interactive system architecture studies, even with a low level of detail in the system architecture description. This enables a system architect to already consider many candidate architectures simultaneously from a safety perspective as well as from an aircraft-level impact point of view, thereby supporting safety informed decision-making.

Specification of a notional fuel system architecture

This section demonstrates the ability to specify and represent aircraft fuel system architectures using the generic element descriptor. It also shows how the ASSESS L1-M1 quantitative safety assessment tool can be used to evaluate these architectures. The study presented here develops a notional fuel system architecture for the EcoJet based on fuel system specification rules developed by [175]. The notional architecture is then evaluated using the graph2AltaRica tool.

Based on a review of the fuel systems of different aircraft, Rodriguez [175, p. 20] breaks down the fuel system into four subsystems. These are the engine feed, fuel transfer, fuel quantity & indication and tank venting subsystems. This study focuses on specifying a notional architecture for the engine feed subsystem.

Rodriguez also proposes rules pertaining to the architecture of the engine feed subsystem. These rules are proposed based on existing architectures and to ensure compliance with FAR Part 25.991 [176] and FAR Part 23.991 [177] regulations that are applicable to turbine-powered aircraft. The rules listed below specify the number of tanks and the number of fuel feed pumps that are required in the engine feed subsystem.

1. The number of tanks must be equal to or greater than the number of engines to facilitate fuel system independence [175, p. 23, line 25]

2. At least two fuel pumps are required per engine [175, p. 23, line 33]

Based on these rules, a notional[5] fuel feed system architecture is specified for the Bombardier EcoJet, as shown in figure 4.18. A wing tank is provided on each wing alongside a center tank. An auxiliary tank is provisioned in the aft section to supply the APU. Cross-feed valves provide the ability to transfer fuel between the wing tanks, though the fuel transfer subsystem is not the focus of this study, and its representation is merely incidental.

[5] The architecture shown here is developed based on safety rules and simplifying assumptions regarding the fuel tank arrangement and APU location, which do not reflect the actual fuel system architecture of the EcoJet.

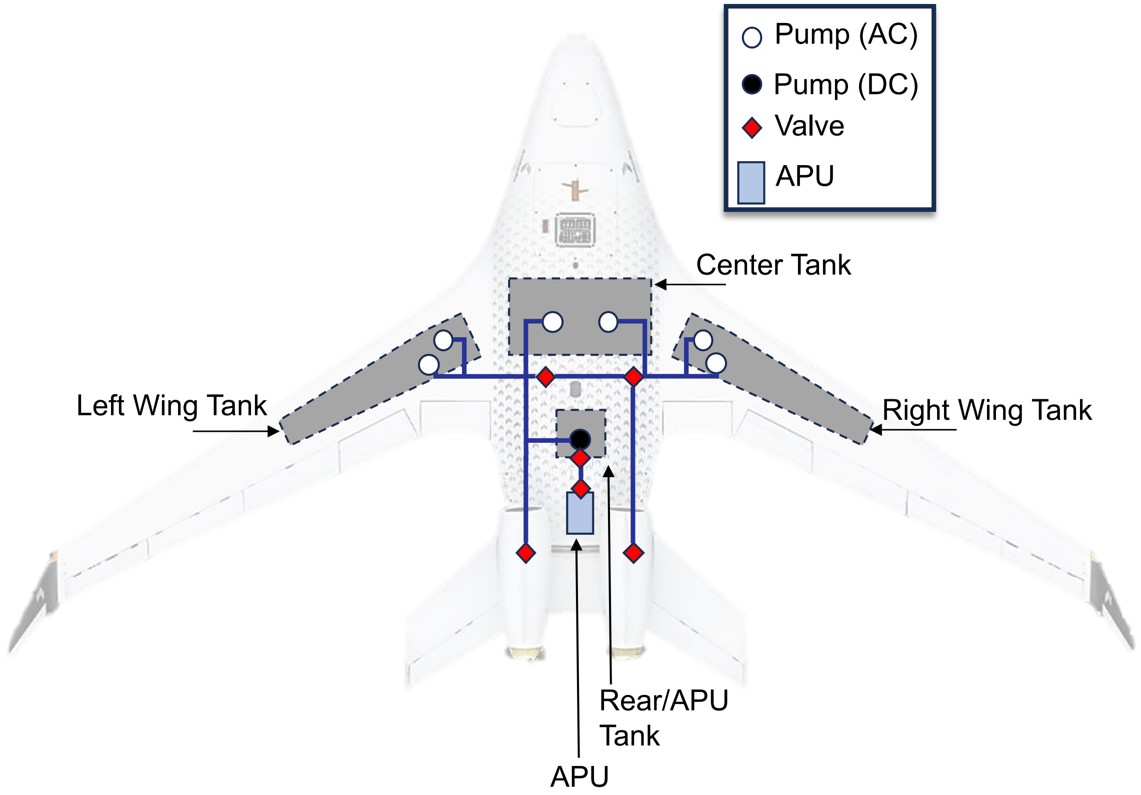


Figure 4.18: Overview of notional fuel-feed subsystem architecture for the EcoJet

The fuel-feed subsystem architecture shown in figure 4.18 is then represented using the generic element descriptor as shown in figure 4.19, which represents the power view. Since the fuel pumps require electrical power, the source elements represent engine-based electrical sources. The distribution elements are electrical distribution systems, the consumer elements represent fuel pumps, and the energy storage elements represent fuel tanks. The generic element descriptor is enriched with component information, such as power type and failure rate, which can be used to develop an FTA using the Graph2AltaRica tool.

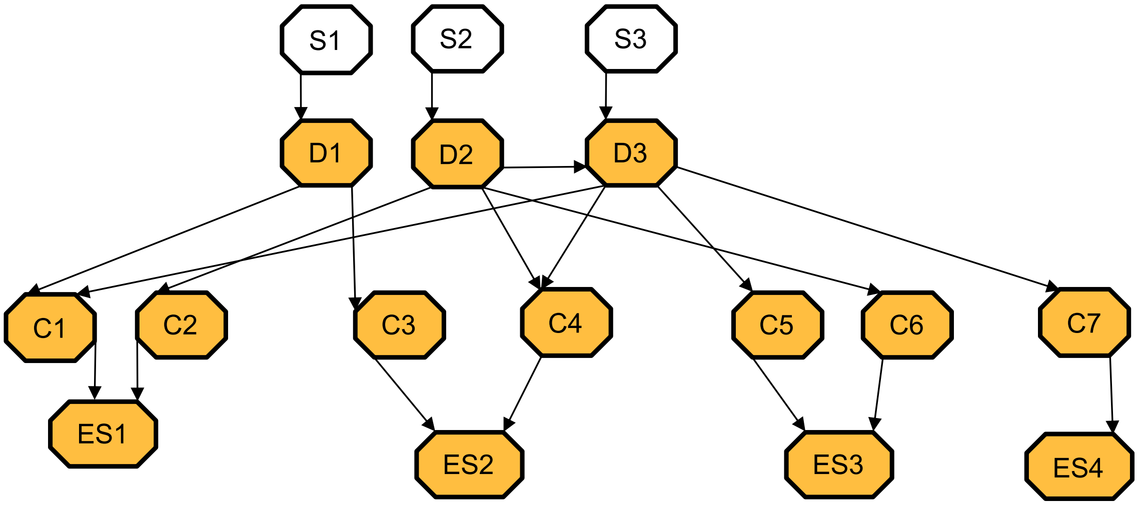


Figure 4.19: Generic element description of the power view of a notional engine feed subsystem architecture

Different failure conditions, such as fuel pump failure or engine failure leading to the loss of the engine fuel feed function, can be evaluated. This early estimation of safety characteristics can help inform the number of fuel pumps, tanks and other subsystem components which directly impact the weight and aircraft-level parameters, thereby providing an early quantification of both safety and the impact of the system architecture on aircraft metrics.

Note on fuel flow view for novel hydrogen-based aircraft concepts

Though not shown in this section, the fuel flow view can also be used to represent the fuel system architecture for unconventional aircraft, such as novel hydrogen-fuelled aircraft concepts. In a collaborative project, Kuelper et al. [178] have used the framework presented in this thesis to develop both power and fuel system views of a novel hydrogen-based aircraft's fuel system architecture. The Graph2AltaRica tool was used to evaluate failure cases and to determine which fuel system architectures met safety requirements. Furthermore, the compatibility of the generic element descriptor with other system architecting frameworks and MBSE environments was also demonstrated by specifying an architecture in MATLAB System Composer and converting it to both power and fuel views using the generic element descriptor, thus creating a direct link between MBSE and MBSA.

4.3.4 Case Study 7: Electric Vertical Take-off & Landing Aircraft (e-VTOL)

e-VTOL aircraft concepts have emerged for Urban Air Mobility (UAM) applications. Several manufacturers, such as Joby Aviation, Beta Technologies and Lillium, are developing aircraft that are currently on the path to certification. These aircraft present a novel integration of functions such as lift generation, thrust generation and flight control. The combination of different functions allows for the definition of novel electrical power system architectures to provide power to elements that fulfill the thrust, lift, flight controls or other shared safety-critical functions. The configuration of the eVTOL aircraft also plays an important role in its safety characteristics. The number of electric motor-driven propellers, their location, and the changes in configuration between different phases of flight, such as cruise and hover, can all have an impact on the system architecture and overall safety characteristics of the aircraft.

As a result, given an initial configuration of an e-VTOL aircraft, an early representation and safety analysis can help provide insight into the overall safety characteristics of the system architecture. This section will demonstrate how the generic element descriptor can represent the system architectures of the novel aircraft, highlight the visual analysis of the architecture that the graph-based generic descriptor enables and show that the generic safety rules outlined in this section also apply to a novel e-VTOL power system architecture. Finally, the utility of early FTA generation will be shown in evaluating e-VTOL potential e-VTOL power system architectures. The e-VTOL aircraft used in this study is based on the concept outlined in a patent filed by Joby Aviation, as shown in figure 4.20. The subsequent section will address the following topics:

1. Representation of e-VTOL power system architectures using the generic element descriptor

2. Demonstration of early FTA generation as a decision-making tool for the analysis of novel e-VTOL architectures

Preliminaries on e-VTOL certification

Before examining the safety characteristics of the e-VTOL power system architectures, it is important to understand the certification regulations to which these aircraft are subject. The FAA classifies e-VTOLs as "powered-lift" aircraft that exhibit characteristics of both a rotor-craft and an airplane. Since airworthiness standards for "powered-lift" category aircraft have not been established, "powered-lift" aircraft fall under special class aircraft. Under 14 CFR 21.17 (b) [179], special class aircraft are subject to 14 CFR parts 23, 25, 27 [180], 29 [181], 31 [182], and part 35 [183] where applicable and in addition to any other airworthiness criteria the FAA deems relevant. In addition to the aforementioned certification criteria, the FAA has prescribed additional criteria for the Joby Model JAS4-1 aircraft. These include expectations for the aircraft to be capable of a controlled emergency landing in the event of a loss of thrust required for continued safe flight (JS4.2105 -f & -g) [184], criteria related to aeromechanical stability and other considerations including flight controls and lightning among others.

Additionally, regulations such as JS4.2405 [184] prescribe that no single failure of the thrust or power control system should prevent continued safe flight and landing of the aircraft. Similarly, JS4.2430 [184] prescribes the need to establish independence between multiple energy storage and supply systems aboard the aircraft such that no single failure of a component on one system will affect the functioning of the other. The installation of equipment and systems is subject to the stipulations of 23.2510. Similarly, the regulations pertaining to systems power generation and storage are those prescribed in 23.2525 [185]. A system safety assessment is required to comply with JS4.2710 -g and JS4.2733 - h [184], and depending on the applicable means of compliance, a safety assessment using the ASTM-F3230-24 [186] may also be required to comply with Part 23.2510. Thus the ability to model and evaluate the propulsion, electrical storage and distribution systems of e-VTOL aircraft in conceptual design can be crucial in establishing the overall feasibility and potential certifiability of an e-VTOL concept.

Modelling the propulsion and electrical distribution architecture of an e-VTOL

The e-VTOL concept shown in figure 4.20 a) is based on information from a patent filing by Joby Aviation [187] and a publicly available 3D CAD model of the Joby S4 aircraft [188]. In this study, the presented e-VTOL concept and the specification of its system architecture are treated as notional. Any safety insights garnered from this study are restricted to the concept presented here and subject to simplifying assumptions made in the absence of detailed information.

The e-VTOL concept shown in figure 4.20 a), features six propulsors that provide thrust during cruise, lift during take-off and hover and also provide lift and thrust during the transition to and from hover to cruise. The fixed-wing provides lift during cruise and is equipped with control surfaces for attitude control during cruise flight. The information provided in [187, p. 24] shows a power system with four batteries that power six electric motors. Another schematic [187, p. 26] shows the various flight control surfaces and the power supplied by the batteries to actuate these surfaces. Figure 4.20 b) shows the installation of four battery packs in two of the fore-propulsor pylons and two in the wings of the aircraft, respectively.

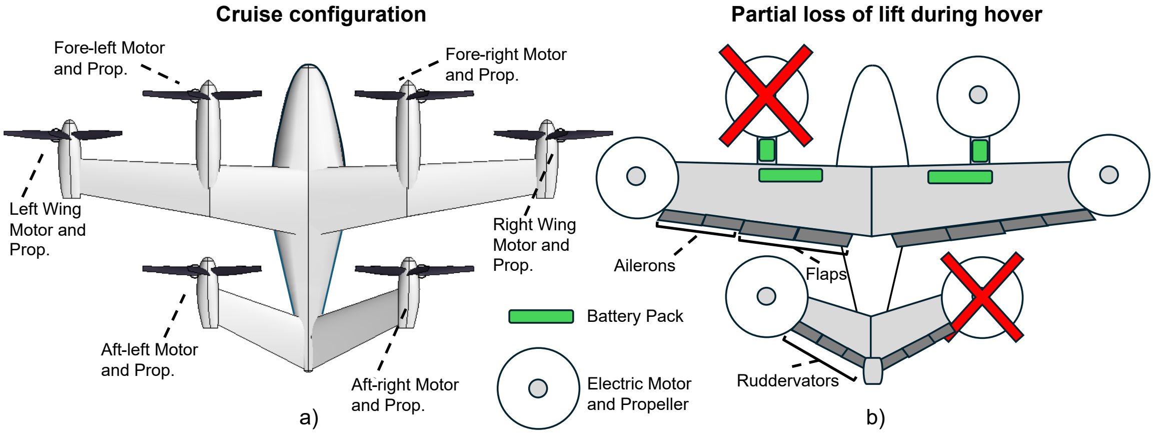

Cruise configuration
Fore-left Motor and Prop.
Fore-right Motor and Prop.
Left Wing Motor and Prop.
Right Wing Motor and Prop.
Aft-left Motor and Prop.
Aft-right Motor and Prop.
a)
Partial loss of lift during hover
Ailerons
Flaps
Battery Pack
Electric Motor and Propeller
Ruddervators
b)

Figure 4.20: Notional e-VTOL concept aircraft based on the Joby S4

A single actuator is assumed to be assigned to each surface. On the propulsion side, it appears that there are two electric power distributions and that each electric motor is dual-wound and is able to receive power from two different sources. It is not entirely clear if the power switch represents an electrical distribution assembly or if each connection to an electric motor is through a dedicated electric bus. Therefore, both options are modelled and shown in figure 4.21 and figure 4.22, respectively.

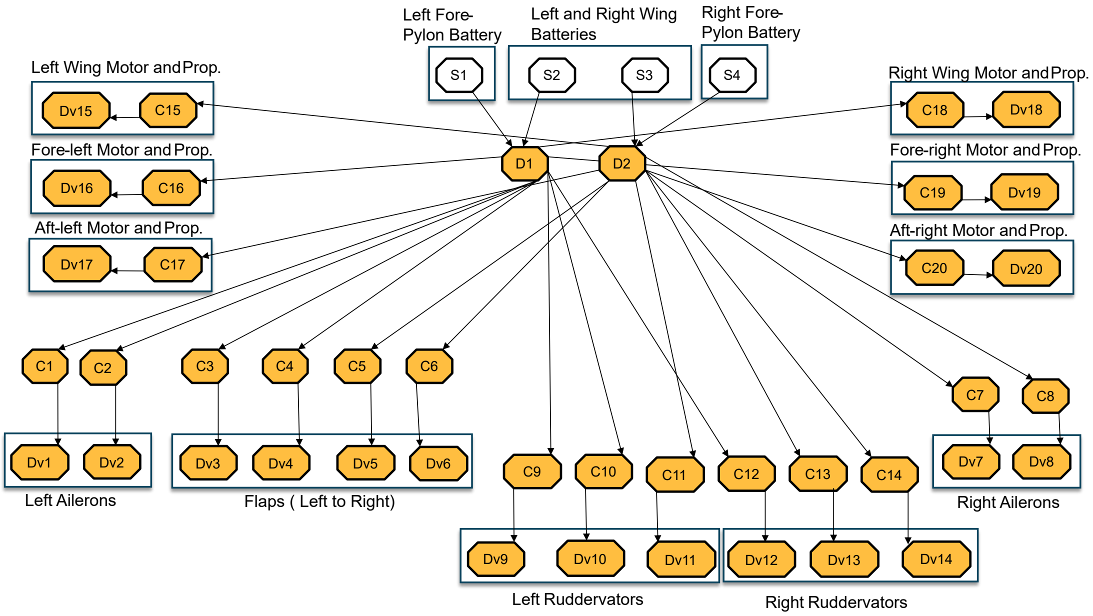


Figure 4.21: Architecture 1: Power view of a notional electrical distribution architecture for an e-VTOL aircraft

The combination of four batteries and six dual-wound electrical motors is claimed to be capable of powering an emergency hover maneuver with four out of six electric motors operational. Figure 4.21, which represents the power view of a notional electrical distribution architecture which has two main distribution systems through which power is routed from the four main batteries. Each electric motor receives two power inputs that are sourced from a pair of main batteries. The actuators for the flight control surfaces also receive power from the two electrical distributions, D1 and D2.

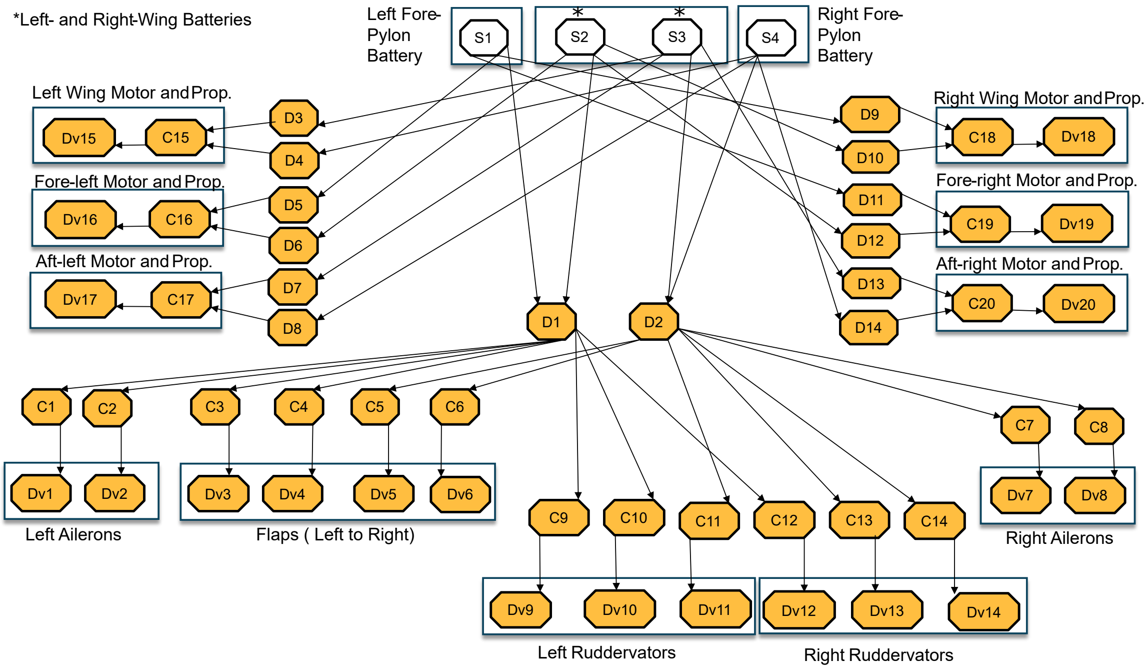


Figure 4.22: Architecture 2: Power view of a variant of the notional electrical distribution architecture for an e-VTOL aircraft shown in figure 4.21

In figure 4.22, a variant of architecture 1 from figure 4.21 is shown where the distribution elements D1 and D2 used solely to power the control surface actuators, and each electric motor has two dedicated distributions that power one winding each. A control architecture featuring three flight control computers powered by electric buses can also be specified but is not shown here as the power allocation to the control elements is not in the scope of this study. Each architecture is then evaluated using the partial loss of thrust during hover due to the loss of the fore-left and aft-right electric motor. Another failure condition that is evaluated is the loss of fore-left, fore-right and aft-right electric motors.

The failure condition evaluated in this study is for illustrative purposes only. A formal FHA would be required to ascertain the overall criticality of the associated functional failure. Here, it is assumed that the criticality of this functional failure is classified as catastrophic. The intent is to show that a complex architecture such as an e-VTOL's electrical power system can be modelled with relative convenience at the conceptual design phase, and the results provide meaningful insight with which to make further architectural decisions. Table 4.14 shows the results of the evaluation of the safety model generated from the generic element descriptor using the Graph2AltaRica tool.

Table 4.14: Overview of evaluated failure rates for architecture 1 and architecture 2

Architecture ID	Partial loss of lift	
	Loss of fore-left and aft-right electric motors (per FH)	Loss of fore-left , fore-right and aft-right electric motors (per FH)
Architecture 1	4.10E-09	3.46E-09
Architecture 2	1.02E-12	1.02E-12

The results show that both architectures meet the safety requirements. Architecture 1 exhibits a minimal cutset that is the loss of both D1 and D2 while architecture 2 has the cutset with the loss of consumer elements C16 and C20 at the highest importance due to the increased redundancy in distribution systems. Architecture 2 exceeds the safety requirement, and therefore, architecture 1 is deemed sufficient for this e-VTOL concept.

Overall, this study successfully demonstrates the ease of modelling and evaluation that comes with using the generic element descriptor and the utility of the automatic transfer to a safety model that enables a rapid evaluation of this unconventional system architecture.

4.3.5 Case Study 8: Conceptual Zonal Safety and Particular Risk Assessment for the NASA SUbSonic Aft-mounted TurbofAN (SUSAN) Aircraft

The objective of the safety-focused systems architecting framework is to enable safety assessment during conceptual design. It specifically targets enabling safety analysis to evaluate the impact of safety considerations on the overall feasibility of an aircraft concept. An important aspect in establishing overall feasibility is ensuring that safety risks are considered and tested by installing the components inside the aircraft. Identifying safety risks before the configuration is frozen can give the conceptual designer the flexibility to modify the system architecture or even the aircraft configuration and can prevent the need for extensive rework in later design stages.

The safety assessment process specified in the SAE ARP4761 includes Particular and Zonal Risk Analysis. Some important particular risks are uncontained engine rotor failure (UERF), APU Rotor Failure, Ram Air Turbine blade release and propeller blade release, among others. The UERF particular risk is analyzed to minimize the hazard in the likelihood that high energy fragments from such a failure should impact critical systems equipment. Furthermore, these risks affect the placement of system components for which redressal in conceptual design is beneficial.

The advisory circular AC 20-128A [189] sets acceptable means of compliance for meeting certification requirements FAR 25.901 [190] and 25.903 [191] (d)(1), for Part 25 aircraft. These regulations concern the installation of the powerplant and the requirement that no single failure or combination of failures will jeopardize the safe application of the aircraft. Part 25.901 (b) [190] specifies that power plant installation complies with 14 CFR 33.5, which provides further engine installation instructions.

The suggested design considerations to mitigate the risk of system components being impacted by high energy fragments in a UERF focus on engine placement relative to critical components such as wiring, hydraulic systems and control cables. Another recommendation of the advisory circular is to locate critical components outside of areas susceptible to impact from debris. Protection and shielding of system components, either using supplemental shielding or using existing airframe structure, is yet another strategy to mitigate UERF risk. Reserving space for dry bays to contain fuel leaks caused by potential high-energy fragments is also recommended design guidance.

This case study examines the propulsion and electrical system architecture of the NASA SUbsonic Single Aft eNgine (SUSAN) Electrofan concept [192] using the ASSESSL1-M2 module and provides recommendations on system installations. The SUSAN concept is still undergoing trade-space evaluation, and further developments in the propulsion and electrical system architecture are expected in the future.

The objective of this case study is to demonstrate how the ASSESS L1-M2 can be used to provide early guidance and recommendations on system component installations. The installation recommendations presented here are based on evaluating the notional installation of electrical components subject to particular risk analysis, such as UERF, propeller blade release and wheel rim release. Furthermore, recommendations are also made for identifying potential regions with zonal safety risks by demarcating stay-out zones and dry bays and specifying minimum separation guidelines for system components. The system architecture of the propulsion and secondary power generation systems is obtained from [193], and the aircraft's three-dimensional geometry with the installed generators and a sample of the electrical buses is based on the model shown in [194].

The SUSAN Electrofan concept shown in figure 4.23 is positioned as a large regional jet in terms of range while maintaining a size typical of a single-aisle aircraft. It carries 180 passengers for a design range of 2500 nmi and an economical operation range of 750 nmi [192] at a cruise speed of Mach 0.785. The concept features an aft ducted turbofan and eight ducted fans distributed under each wing.

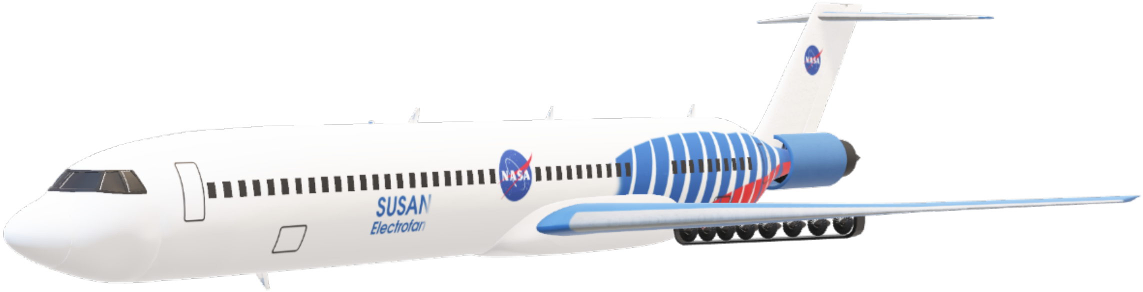


Figure 4.23: NASA SUbSonic Aft-mounted TurbofAN (SUSAN) aircraft concept

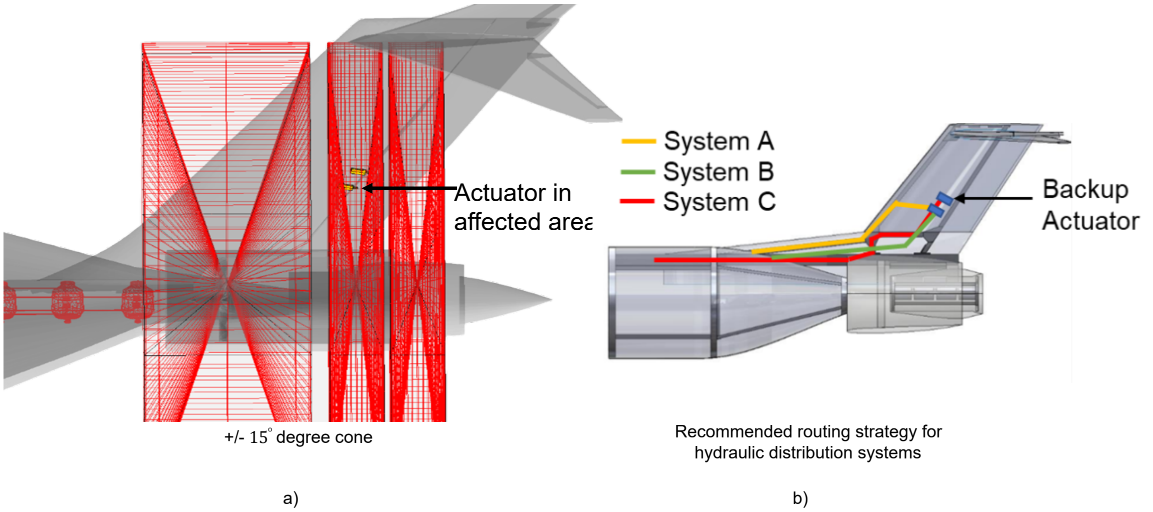
The aircraft uses hybrid electric propulsion and features 20 Megawatts (MW) of electrical power generation from the turbofan using four generators. In addition to the generators, the aircraft has two batteries to provide electrical power backup in the case of engine or generator failure. A single-use battery provides power to the distributed electric propulsors for up to 30 minutes in an engine or generator out scenario, while sixteen smaller batteries serve a dual purpose of both providing electrical power to the motors and supplying loads related to the Turbine Electrified Energy Management Concept described in [195].

The electrical power from the four generators is distributed using a direct bus architecture to the distributed ducted fans such that each fan is powered by its own individual electrical bus. This choice requires each generator to have four outputs with each one feeding a specific electric motor. The buses from each generator are distributed symmetrically to the electric motors powering the ducted fans. The concept uses an AC electrical distribution system and supplies the motors with AC power.

It is important to note that the system architecture presented here is preliminary, and the SUSAN concept and its architecture are under development. The purpose of this section is to demonstrate the utility of the ASSESS L2-M2 module in conducting conceptual ZSA and PRA to aid in safety-driven decision-making during aircraft conceptual design. The results presented in this section are based on notional assumptions of system placement and, unless otherwise cited, do not represent the specifications of the SUSAN Electrofan team at NASA.

4.3.5.1 Examining a case of Uncontained Engine Rotor Failure (UERF) on the rudder actuation system

The SUSAN concept is positioned as a regional jet but also exhibits characteristics of a single-aisle aircraft. The baseline for comparison in initial studies is specified as being a 737-800-like aircraft. The preliminary concept even specifies the required engine thrust to be similar to that of a Boeing 737-800 aircraft. As a result, this study also adopts the rudder actuation system from the 737-800 aircraft. This system consists of powered yaw actuation provided by a dual actuator as a primary actuator which is supplied by two different hydraulic systems. A second backup actuator is supplied by an independent third hydraulic system. This hydraulic system is pressurized by an electric motor pump, has its own reservoir, and is assumed to be situated in the aircraft aft-equipment bay. In this analysis, the region affected by a UERF stemming from both the fan and the first and last turbine stages is considered. The affected region or the rotor burst zone for the fan blade is modelled as a conical region with a spread of +- 15 degrees. The first and last turbine stages consider a spread of +- 5 degrees. Figure 4.24 a) shows the actuators the rotor-burst zones (coloured red) overlapping the actuators (coloured yellow). Though the placement and size of the actuators are notional, the analysis shows that significant parts of the rudder, including potential actuator placement locations, are within at least one rotor burst zone. However, there are two regions on either side of the turbine rotor burst zones that remain unaffected.



+/- 15° degree cone

Recommended routing strategy for hydraulic distribution systems

Figure 4.24: Overview of affected areas for a UERF case along with recommended routing and shielding strategy for hydraulic distribution systems

Other factors associated with the actuators being in the rotor burst zone are the mechanical control cables and the hydraulic power supplies. AC 20-128A [189, p. 21] adopt conservative assumptions for the effects of components inside the affected zones. Control cables struck by rotor fragments are assumed to become disconnected. Hydraulically actuated surfaces that do not have a fail-safe setting are assumed to fail and float. As a result, the routing of control cables and hydraulic supplies needs to be carefully considered.

A recommended design practice here is to provide protection to the different hydraulic supplies by routing them through or behind the airframe structure. In this case the tail cone/fuselage mount shown in figure 4.24 b), could provide protection to two sets of hydraulic supply and return lines. Supplemental shielding can be provided to the set of supply and return lines for the third hydraulic system. If one of the sets of hydraulic lines is shielded by routing it through the pylon, then a minimum separation distance equal to half of the blade fragment size is recommended between the other sets of hydraulic lines. Here, the prescribed separation holds among sets of pressure and supply lines as well as in between the pressure and supply lines in each set.

The other option is to move the actuators aft of the first turbine rotor burst region since both rotor burst events aren't assumed to occur at the same time. However, this choice of actuator location could also affect control authority and lead to increased requirements on individual actuators. Finally, some more radical suggestions would be to either move the engine/tail cone, reduce the rudder sweep or change the position of the rudder. These will undoubtedly have a significant effect on the concept aircraft and may require resizing.

4.3.5.2 Conceptual ZSA

Understanding the implications of system installations on system safety at early design stages can help in determining the feasibility of an aircraft configuration. The zonal safety assessment helps identify the potential safety risks that system components may face when installed in the aircraft and is typically performed in the PSSA or SSA stages when the system architecture is well-defined. Therefore, a promising system architecture that performs well in metrics such as weight and fuel burn may have the potential to be susceptible to safety risks when installed in an aircraft depending on the characteristics of the zone in which it is installed. These may include environmental factors such as heat, humidity, vibration and fire susceptibility of the zone or even the types of components such as hydraulic lines or fuel lines placed in adjacent zones. Thus, performing a ZSA at the conceptual design stage can help identify system installation strategies and identify any emergent zonal safety risks for further analysis. The Conceptual Zonal Safety Assessment (CZSA) methodology developed by Bamrah et al. [196] is implemented in the ASSESS L2-M2 module and is based on installation guidelines, best practices and uses a weighted scoring approach to determine the overall risk that a specific component faces.

The SUSAN Electrofan concept is still undergoing systems trade studies, and the systems descriptions published in the literature are preliminary. However, the description of the aircraft configuration and the conceptual outline of the systems is an ideal place to demonstrate the application of early zonal safety and particular risk assessment methods. An interesting system architecting case on the SUSAN aircraft is that of the routing of fuel lines to the engines. Since the engines are aft-mounted, the fuel lines need to be routed through the aft equipment bay, which contains high-voltage electrical generators, electrical buses and other electrical equipment. Furthermore, if the rudder actuation system on the SUSAN aircraft is assumed to be similar to that of the Boeing-787 − 800, an aircraft against which the SUSAN aircraft is benchmarked, then there is a potential for zonal risks as hydraulic manifolds and reservoirs cannot be placed in zones with potential risk of fire.

The aft equipment bay of the SUSAN aircraft is assumed to begin immediately behind the aft-pressure bulkhead. The aft equipment bay can be considered as a single zone containing the generators and associated equipment, such as electrical buses and power converters. The fuel lines and hydraulic components associated with hydraulic system C are either situated in this zone or pass through it. Another zoning strategy would be to divide the aft-equipment bay medially along the longitudinal axis into two zones. The upper zone could house the hydraulics and control cables for the actuators and the fuel lines. Alternatively, the fuel lines could be routed through the lower zone, and the hydraulics could be maintained in the upper zone. Finally, a third strategy could be considered where the fuel lines and control cables are routed through the upper zone and the hydraulic manifolds and reservoir are kept in the lower zone while the hydraulic lines pass through the upper zone.

The three strategies are analyzed using the CZSA to develop metrics for comparison based on installation guidelines and risk scoring using the known characteristics of the zone environment. Figure 4.25 illustrates all three strategies evaluated in this study.

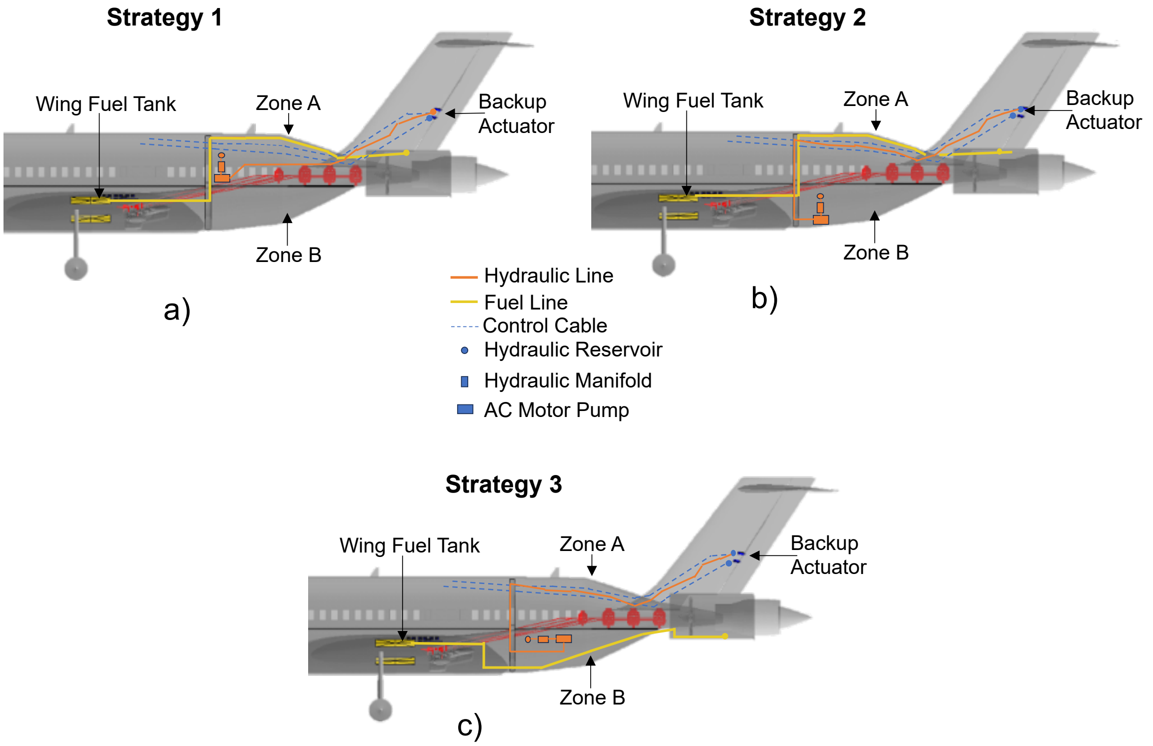


Figure 4.25: Overview of three different system component placement strategies subject to a ZSA

The main components installed within or passing through each zone are the fuel line, hydraulic line, mechanical control cables and the components of the hydraulic system such as hydraulic reservoir, manifold and AC motor pump that pressurize hydraulic system C which is connected to the backup actuator. The hydraulic system components are modelled as primitive shapes similar to how they are represented in similar studies pertaining to zonal safety and thermal risk performed by Bamrah et al. [196, 197], and Sanchez et al. [198] respectively. Here, these abstract representations are shown only to help visualize each strategy. Fuel lines are coloured yellow, and control cables are represented as dotted lines.

Strategy 1, shown in figure 4.25 a), places the hydraulic system components in zone A and routes the control cables and hydraulic lines through the same zone. Several options are also considered while evaluating strategy 1. The first option is to specify the zone as not being under any temperature or pressure control and having a medium packing density. The second option sets the zone as being temperature and pressure-controlled, while the third option considers a low packing density. Packing density is defined by Bamrah et.al as the ratio of the sum of component volume to the overall zone volume. The inputs for the CZA include the definition of each zone and the allocation of components to the zone. Furthermore, the intrinsic risks of each component must also be defined. For the components considered in this study, the intrinsic risks are those pertaining to whether the components carry hazardous, flammable or pressurized fluids that are susceptible to leakage. In this case, the fuel lines, hydraulic reservoir, hydraulic manifolds, AC motor pump and hydraulic lines are given the highest ranking for susceptibility to fluid leakage.

Another aspect of the component that needs to be specified is its operating temperature in relation to the average temperature of the bay. Here, a conservative approach is taken by assuming that the operating temperature of the component is higher than the average temperature of the zone. The intrinsic risks associated with the electrical components are flagged since the power output of the generators is in the megawatt range and the electrical wiring is at a high voltage level. Finally, the flammability and presence of moving parts are assessed to determine the level of risk associated with these characteristics. For example, the AC motor pump is set at a higher risk level for the presence of moving parts as compared to the hydraulic manifold or hydraulic reservoir. Next, the characteristics of the zone environment are specified. In strategy 1, the zone is assumed to lack any temperature and pressure control. It is assumed that the zone will be subject to significant vibration due to the proximity to the engine and the placement of prime movers within the zone. The zone is assumed to be ventilated and has provision for fluid drainage. The zone is assumed to have a medium packing density ranging between 11 and 30 percent.

The zone is also evaluated for susceptibility to particular risks such as lightning strikes, bird strikes and oxygen hazards and is assigned a low score representing low susceptibility to the aforementioned hazards. However, the zone could be classified as a fire zone according to the 14 CFR 25.1181 [199], which identifies engine accessory zones as such. Since the generators are assumed to be in this zone they can be considered to be an engine accessory and the requirements of 14 CFR 25.1181 (b) and 14 CFR 25.863 [200] (a) & (b-3) require that such a zone must be provisioned with means to minimize the risk of ignition of flammable fluids or vapours and to minimize the resultant hazards if ignition does occur. The zone is assumed to also be susceptible to the leakage of flammable fluids due to the presence of a hydraulic reservoir and fuel lines.

Strategies 2 and 3, as shown in figure 4.25 a) and b) respectively, modify the placement and routing of the hydraulic system and fuel line. Strategy 2 places the hydraulic system in Zone B, which is below the floor, while maintaining the same routing of fuel lines as in Strategy 1 . Strategy 3 routes the fuel lines through Zone B while also placing the hydraulic system components in the same zone. The resulting component risk scores are calculated for each component, based on the zone in which it is installed, the adjacent zones and other components nearby and shown in table 4.15.

Strategy 1 results in a high component overall risk for the fuel line, hydraulic manifold

Table 4.15: Risk quantification for components

Component Name	Strategy 1		Strategy 2		Strategy 3	
	COR	Risk Level	COR	Risk Level	COR	Risk Level
Hydraulic manifold and pipes	0.51	High	0.47	Medium	0.40	Medium
Hydraulic Reservoir	1.00	Very High	1.00	Very High	1.00	Very High
AC motor Pump	0.90	Very High	0.81	Very High	0.72	High
Fuel line	0.51	High	0.47	Medium	0.40	Medium
Electrical wiring	0.26	Medium	0.23	Low	0.21	Low
Generators	0.56	High	0.49	Medium	0.42	Medium
Control cable	0.07	Low	0.06	Low	0.05	Low

and pipes and a very high-risk score for the hydraulic reservoir and ACMP. This is expected as the presence of high-power output electrical generators results in a high-temperature environment. Strategy 2 also results in very high-risk scores for the hydraulic reservoir and ACMP. Placing the ACMP and hydraulic reservoir in a different zone from the electrical generators reduces their respective component risk scores, although the hydraulic manifold and pipes still remain subject to a high-risk environment. Additional routing considerations for the hydraulic lines will, therefore, have to be considered in detailed studies further along in the design process.

Overall, the CZSA and CPRA approaches allow early safety insights by considering the aircraft configuration, system architecture and systems installation together to enable the system architect to make safety-driven design decisions. These early studies can provide additional insight into the overall feasibility of a system architecture at the same time as the development of the aircraft configuration.

4.4 Assessment of Hypotheses

This section reviews and evaluates the validity of the hypotheses introduced in section 2.4 against the presented framework and case studies.

Hypothesis 1 : Identifying the needs of each stage of systems architecting in conceptual design, such as architecture granularity, design traceability, and link to MDAO-based evaluation, can enable the definition of an integrated framework for systems architecting that will incorporate safety assessment. This safety-focused system architecting framework can adapt elements of the traditional safety assessment process to the individual needs of each stage of the systems architecting process and ensure the consistent transfer of architecture information between each step of the process.

Assessment of Hypothesis 1: The safety-focused systems architecting framework introduced in section 3.1 adapts the inherent safety characteristics of systems on certified aircraft by formalizing safety heuristics to evaluate a large design space of architectures. This safety-driven evaluation is enabled by the generic element descriptor that provides sufficient detail to support broad design space exploration as well as individual safety analyses such as fault tree assessment. The descriptor also transfers system architecture information between different architecting stages by enabling a link to MDAO workflows and a formal MBSE environment. Sections 3.3.5 to 3.7.1 demonstrate examples of the application of the proposed framework that supports this hypothesis.

Hypothesis 1.1: A graph-based system architecture descriptor can be used as a consistent medium for system architecture information storage and transfer between system architecture definition and MDAO evaluation as well as to Model-Based system architecture specification.

Assessment of hypothesis 1.1 The generic element descriptor implemented as a network graph is used as a standalone means of representing the system architecture as well as a means of transferring system architecture information between different system architecting activities. These include transfer of information to support system sizing tools in an MDAO workflow (section 3.4.1), the transfer of information to build a formal architecture specification in an MBSE environment (section 3.5) and the use of the graph descriptor to support quantitative safety assessment such as fault tree analysis (section 3.6.2). The successful use of the descriptor to enable system architecture information transfer in each of the above-referenced cases provides evidence to support this hypothesis.

Hypothesis 2: Since the traditional safety assessment process is probabilistic, it tends to drive system architectures to increased redundancy. As a result, safety heuristics for minimum required system redundancy and power allocation developed from existing architectures can be used to filter an ample design space of candidate architectures. This will enable architectures that satisfy basic requirements to be developed further in conceptual design with improved confidence in concept feasibility.

Assessment of Hypothesis 2: Section 3.3 describes the rule-based safety assessment process, at the core of which are the identification and formalization of safety rules. These rules are derived for landing gear braking systems and yaw control actuation systems to demonstrate the application to design space exploration. The case studies presented in this thesis show that a design space can be evaluated for safety, and the rules can be applied using evaluative and generative filtering approaches. Although the rules are linked to technology choices, the representation using generic elements helps identify patterns in power flows from power-generating elements to power-consuming elements that are associated with elements that perform a particular function. This can help derive general rules for minimum required redundancy per function, which can then be applied to novel or unconventional architectures.

Hypothesis 2.1: Reducing the complexity of system architecture description by representing the system architecture using a limited set of abstract components will allow a large design space of architectures to be individually described. These architecture descriptions using the generic elements, when represented using network graphs, provide a semantic framework on which the safety heuristics can be evaluated.

Assessment of Hypothesis 2.1: The generic elements introduced in section 3.3.1 are able to represent three views of a system architecture, i.e., the power, control and mass flow views. The power view has been used to represent the power system architecture for both a landing gear braking system and a yaw control actuation system in sections 3.3.5 and 3.3.6, respectively. The control element has been used to represent the combined control and power view in 3.3.7 and the fuel view in 4.3.3. The use of the generic elements in the form of a network graph allows the rules to be evaluated based on the nature of the connections between graph nodes which represent components of the system. Furthermore, this graph-based representation is also suitable for quantitative safety analysis by evaluating the various paths from source to device elements through the graph as shown in section 3.6.2. Therefore, the use of the graph-based generic element descriptor, which is implemented as a network graph, in the case studies described in this thesis for safety rule evaluation and representation of different system architecture views supports this hypothesis.

Hypothesis 2.2.1: Safety heuristics can be integrated into collaborative MDAO workflows in which the analysis tools are distributed across organizations by extracting and processing the architecture descriptor from a common information exchange schema. The architecture information from the schema is processed directly and tested to see if safety heuristics are met. In cases where the schema is inflexible, the information from the schema is converted to the generic element descriptor, using which the safety heuristics are evaluated.

Hypothesis 2.2.2: Safety heuristics can be integrated into a conventional monolithic MDAO structure using a system architecture descriptor that comprises simplified or generic abstractions of typical system architecture components.

Assessment of Hypothesis 2.2.1 & 2.2.2: In section 3.4, this thesis outlines a generic concept for integrating safety assessment within an MDAO workflow through the system architecture descriptor. The subsequent sections show how information from the architecture descriptor can be managed to integrate safety into an industrial workflow and a collaborative MDAO workflow. The integration of heuristics is shown for a collaborative workflow where the system architecture descriptor was in the form of a CPACS schema. In section 3.4.1, the integration of safety into an industrial workflow is shown by building a generic element descriptor from the information already present in a simpler descriptor associated with the industrial workflow. In both types of workflow, the application examples provided in this thesis support the above hypotheses.

Hypothesis 3: Adopting an MBSE approach for systems architecting in conceptual design by specifying the appropriate granularity in system architecture specification can enable integration within the overall aircraft system engineering process. Furthermore, enriching system specification models with information pertinent to MDAO, extracting information by inference or explicit definition, and transferring information using a suitable medium of information exchange can enable MDAO integration.

Hypothesis 3.1: A Model-Based Systems Engineering approach using an architecture specification framework will enable system architectures to be specified and enriched with MDAO inputs at the appropriate level of granularity to enable system architecture input to an MDAO workflow in conceptual design.

Assessment of Hypothesis 3 & 3.1: In section 3.7 a process for specifying a system architecture model in an MBSE environment, enriching it with system sizing inputs and transferring these inputs to a system sizing tool was shown. The process of information transfer outlined in the aforementioned section supports these hypotheses by enabling a formal architecture specification to be evaluated within an MDAO workflow

Overall, the safety-focused systems architecting framework presented in this thesis enables the integration of elements of safety assessment into each activity of systems architecting, i.e., system architecture definition, representation, and evaluation. It also integrates systems architecting with the formal system engineering process by linking simplified architecture representation models with formal MBSE specifications. The framework supports architecture evaluation both from simplified descriptions of the architecture and from a formal MBSE specification. The applicability of the integrated safety assessment methods supported by simplified architecture representation has been shown to enable the evaluation of the safety characteristics of both conventional and novel aircraft system architectures within the conceptual design stage. The capabilities developed in this thesis will enable safety to be considered at the earliest design stage, i.e., conceptual design and will empower system architects to make safety-driven design decisions when developing the systems architecture of future aircraft.

Chapter 5

Contributions, Conclusions, and Recommendations for Future Work

5.1 Contributions

The contributions of this thesis are listed as follows:

1. Introduces a framework for integrating safety assessment in each stage of systems architecting, i.e., system architecture definition, system architecture representation, and system architecture evaluation, in conceptual design

2. Establishes a unified representation of system architecture using a set of generic architecture elements implemented as a graph-based system architecture descriptor that enables safety to be evaluated in conceptual design

   (a) The unified representation of system architecture using the generic elements creates a link between each systems architecting activity by enabling the transfer of system architecture information at the appropriate level of granularity required by each activity

   (b) The unified representation of system architecture using the generic elements enables safety information about the architecture to be captured and used for qualitative and quantitative safety assessment in conceptual design

   (c) The captured safety information is also transferred between the different system architecting stages as part of the unified description using the generic elements to build a formal MBSE specification that helps support detailed analysis in later design stages

3. Connects MBSE and MDAO within the systems architecting process by deriving relevant system sizing inputs from the system architecture descriptor in both its unified representation form and from a formal MBSE specification model

5.2 Summary

This thesis proposes a safety-focused systems architecting framework for aircraft conceptual design as a means of integrating elements of the formal safety assessment process into the conceptual design stage of aircraft development. The approach proposed in this thesis applies elements of the formal safety assessment process specified in the SAE ARP4761, such as the FHA, ZSA, PRA, and FTA, to each stage of the system architecting process. In doing so, the proposed framework outlines a means of managing the granularity of system architecture by means of a generic element based architecture descriptor to support the exploration of a large design space of system architectures while maintaining a sufficient level of detail to support formal safety assessment methods. The proposed framework also uses the generic element descriptor to serve as a means of information transfer between the system architecture definition, representation and evaluation stages. Furthermore, the descriptor supports the transfer of system architecture information to MDAO workflows and also to a formal MBSE specification.

The framework proposed in this thesis describes how safety rules can be developed, formalized and applied to evaluate the compliance of architectures represented using the generic element descriptor. The framework also supports the evaluation of a large design space of architectures using these rules. The proposed framework also shows how representing the architecture using the descriptor in various views, such as the power, control and mass flow views, helps analyze the safety characteristics of the system architecture in response to critical failures. The use of the power view to identify patterns in power flows in architectures aboard certified aircraft is shown to help derive generic rules that can be applicable to unconventional aircraft system architectures.

The transfer of information between the generic element descriptor and MDAO workflows is also described. Furthermore, methods to use the information stored in the generic element descriptor to support quantitative safety assessment and generate elements such as fault trees are also proposed as part of the presented framework. These quantitative safety methods are also curated according to the system architecting task. Simple methods to evaluate a single quantitative safety metric over a design space of system architectures are accompanied by more detailed analyses of system safety as part of individual architecture safety analysis, envisioned as part of the proposed framework.

The framework also shows how information from the graph descriptor can be used to instantiate a formal specification model in an MBSE environment. A method to conduct a model-driven functional hazard assessment is demonstrated using the formal MBSE specification, and a means of storing the output of the FHA within the model is presented. Finally, each element of the framework is demonstrated using relevant applications based on system architectures of increasing complexity. The aircraft landing gear braking system is used as a simple test case to demonstrate safety rule development, safety-based design space filtering and quantitative safety assessment. The yaw control actuation system is selected as a complex test case to demonstrate rule development, safety filtering, quantitative safety assessment, as well as interactive architecting for unconventional aircraft configurations such as the Bombardier EcoJet and NASA Pegasus aircraft concepts.

5.3 Limitations and Future Work

The framework proposed in this thesis and the accompanying examples support the hypotheses made in section 2.4. However, for each hypothesis, there are some limitations in the supporting studies that need to be addressed to ensure the robustness of the framework. These are discussed below:

5.3.1 Limitations

Hypothesis 1, 1.1 & 2.1

The representation of the power, control, and mass flow views has been demonstrated for the landing gear braking system, flight control actuation system, and fuel systems architecture. Though the descriptor is designed to be extensible to any aircraft system featuring power, signal, and mass flows, demonstration of such using examples for different systems architectures is required for a comprehensive evaluation of its representation capabilities. That being said, the descriptor has shown potential in the modelling of unconventional system architectures such as e-VTOL and unconventional flight control actuation architectures.

Similarly, the safety assessment methods that use the descriptor, such as the rule-based safety assessment and the quantitative safety assessment methods, currently support the flight control actuation system, fuel system and landing gear braking system. The level of detail of the descriptor is currently restricted to that which supports typical architecture evolution in conceptual design. However, the descriptor can support multiple levels of system architecture information and can likely be used to provide architecture inputs to drive higher fidelity analyses such as simulation models.

Hypothesis 2.2.1 & 2.2.2

When the system sizing tool has a descriptor formatted differently or incompatible with the generic element descriptor, a wrapper must be used that converts information between the two formats, or the safety rules must be evaluated with whatever information is available in the native descriptor. The extensibility of the generic element descriptor means that with a relativity limited number of parameters, such as the number of engines, the number of distribution systems of a specific power type and the number of consumer elements that receive a certain power type, it is possible to easily build a generic element representation on which to evaluate safety rules. The implementation of the presented framework is such that the architect can generate a baseline representation using the aforementioned parameters and fill in additional information by specifying the connections between different components and other component-specific information. In this way, the safety rule evaluation can still be carried out in cases where the native architecture description is insufficient.

Hypotheses 3 and 3.1

A formal system architecture specification model is shown in section 3.7 to provide sufficient information to the system sizing tool in a collaborative MDAO workflow. The example shown in this thesis prescribes a means of extracting information. However, this process can benefit from automation to enable efficient information transfer between the architecture specification and the MDAO workflow. Another aspect to consider is the application of properties to elements in the model. The PVMT tool used in this thesis is versatile but can also benefit from further integration with the process of transferring information. Ideally, within the context of the Capella MBSE tool, a custom interface or wizard needs to be defined to allow the automated transfer of information from the model to the descriptor and finally to the system sizing tool.

5.3.2 Axes for Future Work

The potential research axes for future work are listed as follows:

1. Improvement of quantitative safety assessment methods

2. Development of machine learning-based system architecting enabled by the graph-based implementation of the generic element descriptor

Improvement of quantitative safety assessment methods

The quantitative safety methods could be improved by considering latent failures and more sophisticated component behaviours. Additional failure models could also be integrated, and the impact of component installation and the identification of hazardous conditions when performing PRA and ZSA also need to be considered in the definition of failure conditions within the automatically generated safety model.

Development of machine learning-based system architecting enabled by the graph-based implementation of the generic element descriptor

Graph-based descriptions of molecular structures have been used in algorithms for drug discovery in chemo-informatics. Taking a corollary, a similar approach could also be used to train models to recognize patterns in system architectures described using the generic element representation. The enablers for such an approach will be to develop node labelling or prediction models, link prediction models and categorization models. Such studies have been attempted at the aircraft systems lab but have had limited success due to the lack of a diverse set of training data in the form of generic representations of a large and diverse set of system architectures. Therefore, further efforts in developing sufficient training data to enable such studies will also be required.

5.4 List of Publications

The publications listed below document the development of the methods presented in this thesis.

Journal Articles

1. **[J1*]** A. K. Jeyaraj and S. Liscouët-Hanke, "A Safety-Focused System Architecting Framework for the Conceptual Design of Aircraft Systems," *Aerospace*, vol. 9, no. 12, p. 791, 2022.

   My contribution: I developed the framework, wrote and edited the manuscript and implemented the test case.

2. **[J2*]** V. Mohan, A. K. Jeyaraj, and S. Liscouët-Hanke, "Systems integration framework for hybrid-electric commuter and regional aircraft," *Aerospace*, vol. 10, no. 6, 2023.

   My contribution: I conceptualized the link between the system architecture descriptor and the proposed systems integration framework, co-curated the case study, and reviewed the manuscript.

3. **[J3*]** N. Tabesh, A. K. Jeyaraj, S. Liscouët-Hanke, and A. Tamayo, "Integration of the functional hazard assessment within a model-based systems engineering framework," *Journal of Aerospace Information Systems*,2024 21:11, 914-926

   Contribution: I conceptualized the Md-FHA process described in the paper, wrote the first draft, curated the test case, and mentored the MASC student, Nikta Tabesh, through the work.

4. **[J4*]** N. Kuelper, A. K. Jeyaraj, S. Liscouët-Hanke, and F. Thielecke, "Integration of a model-based systems engineering framework with safety assessment for early design phases: A case study for hydrogen-based aircraft fuel system architecting," Results in Engineering, vol. 25, art. no. 104249, 2025, doi:10.1016/j.rineng.2025.104249.

   My contribution: I developed the process and implemented a prototype of the quantitative safety assessment methods derived from information obtained from the generic element graph descriptor. I mentored the summer interns who extended the prototype into the Graph2FTA tool. I developed the link between the two frameworks described in this paper in collaboration with the first author and co-curated the test cases.

Conference Papers

1. **[C1]** A. K. Jeyaraj, N. Tabesh, and S. Liscouët-Hanke, "Connecting model-based systems engineering and multidisciplinary design analysis and optimization for aircraft systems architecting," *AIAA Aviation Forum, 2021, AIAA Paper 2021-3077.*

   My contribution: I conceptualized the process of extracting information from the MBSE specification and transferring it to the collaborative workflow. I wrote the manuscript and curated the test cases.

2. **[C2]** M. Fioriti, C. Cabaleiro, T. Lefebvre, P. D. Vecchia, M. Mandorino, S. Liscouët-Hanke, A. Jeyaraj, G. Donelli, and A. Jungo, "Multidisciplinary design of a more electric regional aircraft including certification constraints," *AIAA Aviation 2022 Forum.*

   My contribution: I conducted a case study on rule-based safety assessment using the landing gear braking system as an example and reviewed the manuscript.

3. **[C3]** A. K. Jeyaraj, J. Bussemaker, S. Liscouët-Hanke, and L. Boggero, "Systems architecting: A practical example of design space modeling and safety-based filtering within the Agile4.0 project," *33rd ICAS Congress, September 2022.*

   My contribution: I conducted a case study on rule-based safety assessment using the landing gear braking system as an example and wrote the manuscript.

4. **[C4]** V. Mohan, A. K. Jeyaraj, and S. Liscouët-Hanke, "Systems integration considerations for hybrid-electric commuter aircraft: Case study for the Do-228," *AIAA SciTech 2023 Forum, 2023.*

   My contribution: I co-curated the test cases and edited the manuscript.

5. **[C5]** G.Licheva, V. Mohan, A. K. Jeyaraj, P. Bamrah, M. Mir, and S. Liscouët-Hanke, "System Integration Study for a Hybrid-Electric Commuter Aircraft Concept with a Solar Auxiliary Power System," *AIAA SciTech 2024 Forum, 2024.*

   My contribution: I co-curated the concept definition, test cases, performed the system architecture definition, safety assessment, wrote and reviewed parts of the manuscript, and mentored students involved in the work.

Key: **[*]** Peer-reviewed **[J]** Journal Article **[C]** Conference Publication

References

[1] International Air Transport Association (IATA), "Net zero carbon 2050 resolution." Fact Sheet, 2021. [Online]. Available: https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet----iata-net-zero-resolution/. Accessed: October 12, 2024.

[2] Transport Canada, "Canada's aviation climate action plan, 2022-2030," 2022. [Online]. Available: https://tc.canada.ca/sites/default/files/2022-11/canada-aviation-climate-action-plan-2022-2030.pdf. Accessed: October 12, 2024.

[3] C. Pornet and A. T. Isikveren, "Conceptual design of hybrid-electric transport aircraft," *Progress in Aerospace Sciences*, vol. 79, pp. 114–135, 11 2015.

[4] K. Abu Salem, G. Palaia, and A. A. Quarta, "Review of hybrid-electric aircraft technologies and designs: Critical analysis and novel solutions," *Progress in Aerospace Sciences*, vol. 141, p. 100924, 2023. Special Issue on Green Aviation.

[5] Airbus, "Hybrid and electric flight," 2024. [Online]. Available: https://www.airbus.com/en/innovation/energy-transition/hybrid-and-electric-flight. Accessed: October 12, 2024.

[6] GE Aerospace, "Electric skies: Boeing joins ge and nasa's hybrid electric flight," 2024. [Online]. Available: https://www.geaerospace.com/news/articles/sustainability-technology/electric-skies-boeing-joins-ge-and-nasas-hybrid-electric-flight. Accessed: October 12, 2024.

[7] Airbus, "Ecopulse hybrid aircraft," 2024. [Online]. Available: https://www.airbus.com/en/innovation/energy-transition/hybrid-and-electric-flight/ecopulse. Accessed: October 12, 2024.

[8] Heart Aerospace, "Es-30 electric aircraft," 2024. [Online]. Available: https://heartaerospace.com/es-30/. Accessed: October 12, 2024.

[9] AGILE 4.0 Project, "AGILE 4.0 – Towards Cyber-Physical Collaborative Aircraft Development." https://www.agile4.eu/. Accessed: 2024-07-16.

[10] J. Gould, "Aviary: A new nasa software platform for aircraft modelling," 2024. [Online]. Available: https://www.nasa.gov/aeronautics/aviary-software-overview/. Accessed: October 12, 2024.

[11] I. van Gent, G. L. Rocca, and M. F. M. Hoogreef, “Cmdows: a proposed new standard to store and exchange mdo systems,” *CEAS Aeronautical Journal*, vol. 9, pp. 607–627, Dec 2018.

[12] L. Boggero, T. Lefebvre, W. J. Vankan, B. Beijer, V. Saluzzi, and B. Nagel, “The agile4.0 mbse-mdao development framework: overview and assessment,” (Stockholm), ICAS2022,, 2022.

[13] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, “OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, April 2019.

[14] M. Alder, E. Moerland, J. Jepsen, and B. Nagel, “Recent advances in establishing a common language for aircraft design with cpacs,” in *Aerospace Europe Conference*, (Bordeaux), 2020.

[15] M. K. Fuchs, F. Beckert, J. Biedermann, and B. Nagel, *Experience of Conceptual Designs and System Interactions for the Aircraft Cabin in Virtual Reality*.

[16] S. J. Altelarrea, *Building safety into the conceptual design of complex systems. An aircraft systems perspective*. Phd thesis, Cranfield University, School of Aerospace, Transport and Manufacturing, Cranfield, UK, June 2021.

[17] M. Bendarkar, *An Integrated Framework to Evaluate Off-Nominal Requirements and Reliability of Novel Aircraft Architectures in Early Design*. Phd thesis, Georgia Institute of Technology, Atlanta, GA, USA, May 2021. [Online]. Available: http://hdl.handle.net/1853/64762.

[18] D. P. Raymer, *Aircraft Design: A Conceptual Approach*. Reston, VA: American Institute of Aeronautics and Astronautics, 6th ed., 2018.

[19] S. Liscouët-Hanke, *A Model Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures*. PhD thesis, Université Toulouse III - Paul Sabatier, 2008. Accessed: 2019-07-16.

[20] M. W. Maier and E. Rechtin, *The Art of Systems Architecting*. CRC Press, 2009.

[21] SAE International, *ARP4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. SAE International, 1996. Accessed: 2019-1-15.

[22] SAE International, *Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment ARP4761A*. SAE International, 2023. Accessed: 2024-07-16.

[23] U.S. Federal Aviation Administration, “System Design and Analysis,” Advisory Circular AC 25.1309-1A, U.S. Department of Transportation, June 1988. Accessed: 2024-07-16.

[24] B. Sarlioglu and C. T. Morris, “More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft,” *IEEE Transactions on Transportation Electrification*, vol. 1, pp. 54–64, 2015.

[25] C. Bauer, K. Lagadec, C. Bès, and M. Mongeau, "Flight control system architecture optimization for fly-by-wire airliners," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1023–1029, 2007.

[26] J. Martinez, I. Bouhali, L. Palladino, V. Idasiak, F. Kratz, J.-Y. Choley, and F. Mhenni, "Accelerating digital transformation through mbse, multi-physics simulation and digital twin in industry," *INCOSE International Symposium*, vol. 34, no. 1, pp. 691–715, 2024.

[27] Airbus, "Digital transformation," 2023. [Online]. Available: https://www.airbus.com/en/innovation/digital-transformation. Accessed: Oct. 17, 2024.

[28] S. Liscouët-Hanke, "Proposal : Mdao-nextgen : Developing next generation multi-disciplinary design, analysis and optimization capabilities for next generation aircraft," 2019. [Online]. Available: https://cognit.ca/en/project/248511.

[29] W. L. Simmons, *A Framework for Decision Support in Systems Architecting*. Phd thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2008. Thesis Supervisor: Edward F. Crawley.

[30] H. A. Simon, *The New Science of Management Decision*. New York: Harper and Brothers, 1960.

[31] H. A. Simon, *The New Science of Management Decision*. Englewood Cliffs, N.J.: Prentice-Hall, 3rd ed., 1977.

[32] A. Jeyaraj, "A Model-Based Systems Engineering Approach for Efficient System Architecture Representation in Conceptual Design: A Case Study for Flight Control Systems," Master's thesis, Concordia University, 2019. [Online]. Available: https://spectrum.library.concordia.ca/985353/1/Jeyaraj_MASc_F2019.pdf. Accessed: 2024-07-16.

[33] S. Liscouët-Hanke, "A simulation framework for aircraft power systems architecting," in *Proceedings of the 26th International Congress of the Aeronautical Sciences (ICAS)*, International Congress of the Aeronautical Sciences, 2008.

[34] F. Zwicky, *Morphological Analysis and Construction*. Wiley Inter-science, 1948.

[35] W. Engler, P. Biltgen, and D. Mavris, *Concept Selection Using an Interactive Reconfigurable Matrix of Alternatives (IRMA)*. American Institute of Aeronautics and Astronautics, 1 2007. doi:10.2514/6.2007-1194.

[36] D. Judt and C. Lawson, "Methodology for automated aircraft systems architecture enumeration and analysis," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics (AIAA), 2012. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2012-5648. DOI: 10.2514/6.2012-5648.

[37] D. M. Judt and C. Lawson, "Development of an automated aircraft subsystem architecture generation and analysis tool," *Engineering Computations*, vol. 33, pp. 1327–1352, July 2016.

[38] SAE International, "ARP4754A: Development of Civil Aircraft and Systems," 2011.

[39] M. Armstrong, "A process for function based architecture - definition and modeling," Master's thesis, School of Aerospace Engineering, April 2008. Master's Thesis.

[40] M. Armstrong, C. de Tenorio, D. Mavris, and E. Garcia, "Function based architecture design space definition and exploration," in *The 26th Congress of ICAS and 8th AIAA ATIO*, American Institute of Aeronautics and Astronautics, September 2008.

[41] S. Liscouët-Hanke, J.-C. Maré, and S. Pufe, "Simulation framework for aircraft power system architecting," *Journal of Aircraft*, vol. 46, pp. 1375–1380, July 2009. [Online]. DOI: https://doi.org/10.2514/1.41304.

[42] T. Lammering, *Integration of aircraft systems into conceptual design synthesis.* PhD thesis, RWTH Aachen University, Aachen, 2014. Summary in German and English; Aachen, Techn. Hochsch., Diss., 2014.

[43] R. Bornholdt, T. Kreitz, and F. Thielecke, "Function-driven design and evaluation of innovative flight controls and power system architectures," *SAE International Journal of Aerospace*, vol. 8, no. 2, pp. 189–197, 2015.

[44] R. Bornholdt, *Systemübergreifende Analyse und Bewertung von Architekturvarianten neuartiger Flugzeugsysteme anhand von Sicherheits- und Betriebsaspekten.* PhD thesis, Technische Universität Hamburg, 2021.

[45] I. Chakraborty and D. N. Mavris, "Integrated assessment of aircraft and novel subsystems architectures in early design," *Journal of Aircraft*, vol. 54, pp. 1268–1282, 2017.

[46] T. Lampl, R. Königsberger, and M. a. Hornung, "Design and evaluation of distributed electric drive architectures for high-lift control systems," in *66. Deutsche Luft- und Raumfahrtkongress* (D. G. für Luft-und Raumfahrt (DGLR), ed.), 2017.

[47] T. Lampl, D. Sauterleute, and M. a. Hornung, "A functional-driven design approach for advanced flight control systems of commercial transport aircraft," in *Proceedings of the 6th International Workshop on Aircraft System Technologies* (F. T. Otto von Estorff, ed.), p. 3–12, 2017.

[48] T. S. Lampl and M. Hornung, *An Integrated Design Approach for Advanced Flight Control Systems with Multifunctional Flight Control Devices.* American Institute of Aeronautics and Astronautics, 2018.

[49] T. Lampl, T. Wolf, and M. Hornung, "Preliminary design of advanced flight control system architectures for commercial transport aircraft," *CEAS Aeronautical Journal*, vol. 10, no. 2, pp. 613–622, 2019.

[50] T. S. Lampl, *Integrated Design of Advanced Flight Control Configurations and System Architectures.* PhD thesis, Technische Universität München, 2021.

[51] J. H. Bussemaker, P. D. Ciampa, and B. Nagel, "System architecture design space exploration: An approach to modeling and optimization," in *AIAA Aviation 2020 Forum*, vol. 1 Part F, pp. 1–22, American Institute of Aeronautics and Astronautics Inc, AIAA, 2020.

[52] J. H. Bussemaker and L. Boggero, "Technologies for enabling system architecture optimization," in *ODAS Symposium*, (Hamburg), June 2022.

[53] C. Ramchandani, M. Maier, and T. McKendree, "Toward a comprehensive architecture representation model," *INCOSE International Symposium*, vol. 5, no. 1, pp. 699–705, 1995.

[54] Airbus, *Airbus A320 Flight Crew Operating Manual (FCOM) - Hydraulics*, 2024. [Online]. Available: https://www.smartcockpit.com/my-aircraft/airbus-a320/#. Accessed: 2024-10-15.

[55] L. Kushner, "Icd-cept-006: Interface control document - conceptual design phase x-57," Tech. Rep. 20230016539, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia, United States, Jan. 2023. Public Use Permitted. Work of the US Gov.

[56] M. Fuchs, Y. Ghanjaoui, J. Abulawi, *et al.*, "Enhancement of the virtual design platform for modeling a functional system architecture of complex cabin systems," *CEAS Aeronautical Journal*, vol. 13, no. 4, pp. 1101–1117, 2022.

[57] NASA, "Nasa systems engineering handbook," Tech. Rep. NASA/SP-2016-6105 Rev2, National Aeronautics and Space Administration, Washington, DC, 2016. Supersedes SP-2007-6105 Rev 1.

[58] A. Poland, V. Domingo, and W. Worrall, "Interface control document between the solar and heliospheric observatory (soho) experimenters operations facility (eof) core system (ecs) and the soho instrumenters," Tech. Rep. 514-3ICD/0193, NASA, October 1995.

[59] J. A. E. A. (JAXA), "Jem payload accommodation handbook - vol. 8 - small satellite deployment interface control document," Tech. Rep. JX-ESPC-101133-B, JAXA, March 2013. Initial Release: March 2013, Revision A: May 2013, Revision B: January 2015.

[60] R. Tsui, D. Davis, and J. Sahlin, "Digital engineering models of complex systems using model-based systems engineering (mbse) from enterprise architecture (ea) to systems of systems (sos) architectures & systems development life cycle (sdlc)," *INCOSE International Symposium*, vol. 28, pp. 760–776, 2018.

[61] K. Vipavetz, T. A. Shull, S. Infeld, and J. Price, "Interface management for a nasa flight project using model-based systems engineering (mbse)," *INCOSE International Symposium*, vol. 26, pp. 1129–1144, 2016.

[62] S. Saunders, "Does a model based systems engineering approach provide real program savings? – lessons learnt." https://www.omgsysml.org/Does_a_MBSE_Approach_Provide_Savings-Lessons_Learnt-Saunders-200111.pdf, 2011. DSTO Model-Based Systems Engineering (MBSE) Symposium, Adelaide, Australia.

[63] A. Salado and P. Wach, "Automatic generation of contractual requirements from mbse artifacts," Tech. Rep. VT-CM-19-197, Acquisition Research Program, Graduate School of Business & Public Policy, Naval Postgraduate School, Sep 2019.

[64] T. Weilkiens, J. G. Lamm, S. Roth, and M. Walker, *Model-Based System Architecture*. Wiley Series in Systems Engineering and Management, Hoboken, NJ: John Wiley & Sons, Inc., second edition ed., 2022.

[65] DLR, "Cpacs." https://www.cpacs.de/. Accessed: 2024-07-17.

[66] V. Mohan, A. K. Jeyaraj, and S. Liscouët-Hanke, "Systems integration framework for hybrid-electric commuter and regional aircraft," *Aerospace*, vol. 10, no. 6, 2023.

[67] G. P. Krupa, "Application of agile model-based systems engineering in aircraft conceptual design," *The Aeronautical Journal*, vol. 123, no. 1268, pp. 1561–1601, 2019.

[68] N. Tabesh, "A model-based system engineering approach to support system architecting activities in early aircraft design," Master's thesis, Concordia University, 2023. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/992378/1/Tabesh_MASc_F2023.pdf.

[69] E. Crawley, B. Cameron, and D. Selva, *System Architecture: Strategy and Product Development for Complex Systems*. Harlow, England: Pearson Education Limited, global edition ed., 2016. Foreword by Norman R. Augustine.

[70] J. Roskam, *Airplane Design: Part I*, vol. 1. Lawrence, Kansas: Roskam Aviation and Engineering Corp., 2015.

[71] L. E. Zeidner, H. M. Reeve, R. Khire, and S. Becz, "Architectural enumeration & evaluation for identification of low-complexity systems," in *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, vol. 3, (Reston, Virginia), September 2010.

[72] S. Becz, A. Pinto, L. E. Zeidner, R. Khire, A. Banaszuk, and H. M. Reeve, "Design system for managing complexity in aerospace systems," in *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, vol. 2, 2010.

[73] A. G. Garriga, P. Govindaraju, S. S. Ponnusamy, N. Cimmino, and L. Mainini, "A modelling framework to support power architecture trade-off studies for more-electric aircraft," *Transportation Research Procedia*, vol. 29, pp. 146–156, January 2018.

[74] A. G. Garriga, L. Mainini, and S. S. Ponnusamy, "A machine learning enabled multi-fidelity platform for the integrated design of aircraft systems," *Journal of Mechanical Design, Transactions of the ASME*, vol. 141, p. 121405, December 2019.

[75] M. Fioriti, C. Cabaleiro, T. Lefebvre, P. D. Vecchia, M. Mandorino, S. Liscouët-Hanke, A. Jeyaraj, G. Donelli, and A. Jungo, "Multidisciplinary design of a more electric regional aircraft including certification constraints," in *AIAA AVIATION 2022 Forum*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2022.

[76] I. Chakraborty, *Subsystem architecture sizing and analysis for aircraft conceptual design*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2015. https://repository.gatech.edu/server/api/core/bitstreams/1268cd5b-0a8a-4d32-8293-1c6a3bff1e10/content.

[77] S. Wakayama and I. Kroo, "The challenge and promise of blended-wing-body optimization," in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, American Institute of Aeronautics and Astronautics, 1998. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1998-4736. DOI: 10.2514/6.1998-4736.

[78] F. Flager and J. Haymaker, "A comparison of multidisciplinary design, analysis, and optimization processes in the building construction and aerospace industries," Tech. Rep. TR188, Stanford University, Center for Integrated Facility Engineering (CIFE), Dec 2009. p. 7.

[79] P. Piperni, A. DeBlois, and R. Henderson, "Development of a multilevel multidisciplinary-optimization capability for an industrial environment," *AIAA Journal*, vol. 51, no. 10, pp. 2335–2352, 2013.

[80] S. Chiesa, G. A. D. Meo, M. Fioriti, G. Medici, and N. Viola, "Astrid–aircraft on board systems sizing and trade-off analysis in initial design," *Research and Education in Aircraft Design–READ*, pp. 1–28, 2012.

[81] M. Jünemann, F. Thielecke, F. Peter, M. Hornung, F. Schültke, and E. Stumpf, "Methodology for design and evaluation of more electric aircraft systems architectures within the avacon project," in *Proceedings of the Deutscher Luft- und Raumfahrtkongress*, 2019.

[82] P. D. Ciampa and B. Nagel, "Agile paradigm: The next generation collaborative mdo for the development of aeronautical systems," *Progress in Aerospace Sciences*, vol. 119, p. 100643, 2020.

[83] M. Fioriti, P. D. Vecchia, G. Donelli, and P. Hansmann, "Assessing the integration of electrified on-board systems in an mdao framework for a small transport aircraft," in *AIAA AVIATION 2021 FORUM*, 2021. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2021-3094. DOI: 10.2514/6.2021-3094.

[84] F. Sanchez, S. Liscouët-hanke, and A. Tfaily, "Improving aircraft conceptual design through parametric cad modellers – a case study for thermal analysis of aircraft systems," *Computers in Industry*, vol. 130, 9 2021.

[85] S. Sélim, S. Liscouët-Hanke, A. Tfaily, A. Butt, and B. Alphonso, "Scoring approach to assess maintenance risk for aircraft systems in conceptual design," *Journal of Aircraft*, vol. 60, no. 5, pp. 1577–1587, 2023.

[86] M. Buonanno, T. Kwasniak, N. Quay, A. Ross, D. Sagan, C. A. Lupp, and B. Boden, *Incorporation of Producibility Considerations into an Aircraft Multidisciplinary Design Optimization Framework*. American Institute of Aeronautics and Astronautics.

[87] I. Chakraborty and D. N. Mavris, "Heuristic definition, evaluation, and impact decomposition of aircraft subsystem architectures," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, 6 2016. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2016-3144. DOI: 10.2514/6.2016-3144.

[88] P. Zimmerman, "DoD Digital Engineering Strategy," in *20th Annual NDIA Systems Engineering Conference*, (Springfield, VA), October 2017. Presentation. [Online]. Available: https://ndia.dtic.mil/wp-content/uploads/2017/systems/Wednesday/Track3/19819_Zimmerman.pptx. Accessed: Nov. 21, 2024.

[89] T. D. West and A. Pyster, "Untangling the digital thread: The challenge and promise of model-based engineering in defense acquisition," *INSIGHT*, vol. 18, no. 2, pp. 45–55, 2015.

[90] Q. Zhang, J. Liu, and X. Chen, "A literature review of the digital thread: Definition, key technologies, and applications," *Systems*, vol. 12, no. 3, 2024. [Online]. Available: https://www.mdpi.com/2079-8954/12/3/70.

[91] INCOSE, "Systems engineering vision 2020." INCOSE-TP-2004-004-02, International Council on Systems Engineering, San Diego, CA, USA, Sep. 2007. [Online]. Available: https://www.scribd.com/document/80508688/Systems-Engineering-Vision-2020-INCOSE-February-2009-20071003-v2-03, 2007. [Accessed: 15th June 2021].

[92] "Types of models." SEBoK. [Online]. Available: https://sebokwiki.org/wiki/Types_of_Models. [Accessed: Mar. 29, 2019].

[93] T. J. Bayer, "Is mbse helping? measuring value on europa clipper," *2018 IEEE Aerospace Conference*, pp. 1–13, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49540521.

[94] T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner, "Early formulation model-centric engineering on nasa's europa mission concept study," *INCOSE International Symposium*, vol. 22, no. 1, pp. 1695–1710, 2012.

[95] T. J. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, and D. Wagner, "Update on the model based systems engineering on the europa mission concept study," *INCOSE International Symposium*, vol. 23, no. 1, pp. 694–709, 2013.

[96] D. Nichols and C. Lin, "Integrated model-centric engineering: The application of mbse at jpl through the life cycle." INCOSE International Workshop, MBSE Workshop, Jan. 26, 2014. [Online]. [Online]. Available: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:06-iw14-mbse_workshop-application_of_mbse_at_jpl_through_the_lifecycle-nichols-lin-final.pdf. Accessed: Mar. 29, 2019.

[97] M. Chami, P. Oggier, O. Naas, and M. Heinz, "Real world application of mbse at bombardier transportation." SWISSED 2015, Swiss Society of Systems Engineering, Zurich, Switzerland, 2015. [Online]. Available: https://www.researchgate.net/publication/299284191_Real_World_Application_of_MBSE_at_Bombardier_Transportation.

[98] S. Liscouët-Hanke and A. Jeyaraj, "A model-based systems engineering approach for efficient flight control system architecture variants modelling in conceptual design," in *International Conference on Recent Advances in Aerospace Actuation Systems and Components*, pp. 34–41, 2018.

[99] P. G. Mathew, S. Liscouet-Hanke, and Y. L. Masson, “Model-based systems engineering methodology for implementing networked aircraft control system on integrated modular avionics – environmental control system case study,” *SAE Technical Papers*, October 2018.

[100] R. Malone, B. Friedland, J. Herrold, and D. Fogarty, “Insights from large scale model based systems engineering at boeing,” *INCOSE International Symposium*, vol. 26, pp. 542–555, July 2016.

[101] P. D. Ciampa, G. L. Rocca, and B. Nagel, *A MBSE Approach to MDAO Systems for the Development of Complex Products*. American Institute of Aeronautics & Astronautics, 2020.

[102] J. Bussemaker and P. Ciampa, “Mbse in architecture design space exploration,” in *Handbook of Model-Based Systems Engineering* (A. Madni, N. Augustine, and M. Sievers, eds.), Springer, Cham, 2022.

[103] R. Cloutier and I. Obiako, “Model-based systems engineering adoption trends 2009-2018 - sebok.” Online, 2019. Available: https://www.sebokwiki.org/wiki/Model-Based$_{S}ystems_{E}ngineering_{A}doption_{T}rends_{2}009 - 2018.[Accessed : 30 - Jan - 2020]$.

[104] D. Huart and O. Olechowski, “Towards a model-based systems lifecycle: Cpcs from design to operations,” in *Proceedings of the 6th International Workshop on Aircraft System Technologies (AST 2017)*, (Hamburg, Germany), Feb 2017.

[105] S. Liscouet-Hanke, B. R. Mohan, P. J. Nelson, C. Lavoie, and S. Dufresne, “Evaluating a model-based systems engineering approach for the conceptual design of advanced aircraft high-lift system architectures,” in *Canadian Aeronautics and Space Institute AERO 2017*, 2017.

[106] M. Chami and J. M. Bruel, “A survey on mbse adoption challenges,” in *Proceedings of the INCOSE EMEA Sector Systems Engineering Conference (INCOSE EMEASEC 2018)*, (Berlin, Germany), pp. 1–16, Nov 2018.

[107] A. K. Jeyaraj, N. Tabesh, S. Liscouët-Hanke, and S. Liscouet-Hanke, “Connecting model-based systems engineering and multidisciplinary design analysis and optimization for aircraft systems architecting,” in *AIAA Aviation Forum*, 2021. AIAA Paper 2021-3077.

[108] O. Lisagor, T. Kelly, and R. Niu, “Model-based safety assessment: Review of the discipline and its challenges,” in *Proceedings of the 9th International Conference on Reliability, Maintainability and Safety*, (New York), pp. 625–632, Institute of Electrical and Electronics Engineers, 2011.

[109] O. Lisagor, A. McDermid, and D. J. Pumfrey, “Towards a practicable process for automated safety analysis,” in *Proceedings of the 24th International System Safety Conference*, (Unionville, VA), pp. 596–607, System Safety Society Publ., 2006.

[110] G. Point and A. Rauzy, “Altarica constraint automata as a description language,” *Journal of European Systems Automation*, vol. 33, no. 8–9, pp. 1033–1052, 1999.

[111] T. Prosvirnova, M. Batteux, P.-A. Brameret, A. Cherfi, T. Friedlhuber, J.-M. Roussel, and A. Rauzy, “The altarica 3.0 project for model-based safety assessment,” *IFAC Proceedings*

Volumes, vol. 46, no. 22, pp. 127–132, 2013. 4th IFAC Workshop on Dependable Control of Discrete Systems.

[112] Y. Papadopoulos and J. A. McDermid, "Hierarchically performed hazard origin and propagation studies," in *Computer Safety, Reliability and Security: 18th International Conference, Proceedings/SAFECOMP* (M. Felici, K. Kanoun, and A. Pasquini, eds.), vol. 99, (Berlin), pp. 139–152, Springer, 1999.

[113] F. Bruno, M. Fioriti, G. Donelli, L. Boggero, P. D. Ciampa, and B. Nagel, "A model-based rams estimation methodology for innovative aircraft on-board systems supporting mdo applications," in *AIAA Aviation Forum*, American Institute of Aeronautics and Astronautics, 2020. AIAA Paper 2020-3151.

[114] S. Gradel, B. Aigner, and E. Stumpf, "Model-based safety assessment for conceptual aircraft systems design," *CEAS Aeronautical Journal*, vol. 13, no. 1, pp. 281–294, 2021.

[115] S. Maitrehenry, S. Metge, Y. Ait-Ameur, and P. Bieber, "Towards model-based functional hazard assessment at aircraft level," in *European Safety and Reliability Conference (ESREL)*, (Bruxelles, Belgium), p. 390, European Safety and Reliability Association, 2011.

[116] E. Villhauer and B. Jenkins, "An integrated model-based approach to system safety and aircraft system architecture development," in *INCOSE International Symposium*, vol. 25, (San Diego, CA), pp. 1373–1387, International Council on Systems Engineering (INCOSE), 2015.

[117] Y. Jiang, N. Bai, H. Yang, H. Zhang, Z. Wang, and X. Liu, "MBSE-Based Functional Hazard Assessment of Civil Aircraft Braking System," in *Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE 2020)*, (New York), pp. 460–464, Institute of Electrical and Electronics Engineers, 2020.

[118] S. Jimeno, A. Molina-Cristobal, A. Riaz, M. D. Guenov, and S. J. Altelarrea, "Incorporating safety in early (airframe) systems design and assessment," in *AIAA SciTech Forum*, 2019. AIAA Paper 2019-0553.

[119] S. Jimeno, A. Riaz, M. D. Guenov, and A. Molina-Cristobal, "Enabling interactive safety and performance trade-offs in early airframe systems design," in *AIAA SciTech Forum*, (Reston, VA), pp. 1–16, American Institute of Aeronautics and Astronautics, 2020. AIAA Paper 2020-0550.

[120] S. Lübbe, M. Schäfer, and O. Bertram, "Coupling of model-based systems engineering and safety analysis in conceptual aircraft system design," in *33rd Congress of the International Council of the Aeronautical Sciences*, (Stockholm, Sweden), 2022.

[121] M. Schäfer, A. Berres, and O. Bertram, "Integrated model-based design and functional hazard assessment with sysml on the example of a shock control bump system," *CEAS Aeronautical Journal*, vol. 14, no. 1, pp. 187–200, 2023.

[122] S. M. Lübbe, M. Schäfer, V. Voth, A. Berres, and O. Bertram, "Interconnections in model-based safety analysis and systems design on the example of a fuel cell thermal management system for commercial aircraft," in *AIAA AVIATION 2023 Forum*, (National Harbor, Maryland, USA), American Institute of Aeronautics and Astronautics, June 2023. doi:10.2514/6.2023-4196.

[123] K. Lai, T. Robert, D. Shindman, and A. Olechowski, "Integrating safety analysis into model-based systems engineering for aircraft systems: A literature review and methodology proposal," *INCOSE International Symposium*, vol. 31, pp. 988–1003, 7 2021.

[124] K. Lai, "A guideline for the implementation of model-based functional hazard assessment and its integration with model-based systems engineering," masters thesis, University of Toronto, Toronto, 2022.

[125] U.S. Federal Aviation Administration, "Part 25.1309 of chapter i, subchapter c of title 14 code of federal regulations," 2021. Accessed: 2024-07-17.

[126] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press, 2016.

[127] K. Baughey, "Functional and logical structures: A systems engineering approach," *SAE Technical Paper*, no. 2011-01-0517, 2011.

[128] S. Kleiner and C. Kramer, "Model based design with systems engineering based on rflp using v6," in *Smart Product Engineering* (M. Abramovici and R. Stark, eds.), pp. 161–175, Springer, Berlin, Heidelberg, 2013.

[129] U.S. Federal Aviation Administration, "14 CFR Part 23 - Airworthiness Standards: Normal Category Airplanes." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23. [Accessed: Dec. 3, 2024].

[130] U.S. Federal Aviation Administration, "14 CFR Part 25 - Airworthiness Standards: Transport Category Airplanes." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25?toc=1. [Accessed: Dec. 3, 2024].

[131] D. F. Finger, C. Braun, and C. Bil, "An initial sizing methodology for hybrid-electric light aircraft," in *Proceedings of the 2018 Aviation Technology, Integration, and Operations Conference*, (Atlanta, Georgia), American Institute of Aeronautics and Astronautics, June 2018.

[132] D. F. Finger, R. de Vries, R. Vos, C. Braun, and C. Bil, "A comparison of hybrid-electric aircraft sizing methods," in *Proceedings of the AIAA Scitech 2020 Forum*, (Orlando, FL, USA), American Institute of Aeronautics and Astronautics, January 2020.

[133] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.

[134] R. Diestel, *Graph Theory*. Berlin/Heidelberg, Germany: Springer Nature, 2017.

[135] Neo4j, Inc., "arrows.app." https://neo4j.com/labs/arrows/. Accessed: 2024-12-02., 2024.

[136] N. Leveson, C. Wilkinson, H. C. Fleming, J. Thomas, and I. Tracy, "A comparison of stpa and the arp 4761 safety assessment process," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 2014. See pages 10, 15 and 61.

[137] R. Sedgewick, *Algorithms in C, Part 5: Graph Algorithms*. Addison Wesley Professional, 3rd ed., 2001.

[138] U.S. Federal Aviation Administration, "14 CFR Part 23 Subpart F Section 23.2510 - Equipment, systems, and installations." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23/subpart-F/section-23.2510. [Accessed: Dec. 3, 2024].

[139] Federal Aviation Administration, "14 CFR Part 25 Subpart B Section 25.147 - Directional and lateral control." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-B/subject-group-ECFR889f3dfd0ada276/section-25.147. [Accessed: Dec. 5, 2024].

[140] U.S. Federal Aviation Administration, "14 CFR Part 25 Subpart D Section 25.671 - Control systems." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-D/subject-group-ECFRe3ac3aa184c8c5a/section-25.671. [Accessed: Dec. 3, 2024].

[141] Federal Aviation Administration, "14 CFR Part 25 Subpart B Section 25.149 - Minimum control speed." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-B/subject-group-ECFR889f3dfd0ada276/section-25.149. [Accessed: Dec. 5, 2024].

[142] E. Godo, "Flight control system with remote electronics," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 2, pp. 13B1–13B1, 2002.

[143] Thales Group, "FlytRise Flight Control Systems by Thales." Online. Available: https://www.thalesgroup.com/en/markets/aerospace/flight-deck-avionics-equipment-functions/flight-control-systems/flytrise-flight, 2024. Accessed: Oct. 4, 2024.

[144] I. van Gent and G. L. Rocca, "Formulation and integration of mdao systems for collaborative design: A graph-based methodological approach," *Aerospace Science and Technology*, vol. 90, pp. 410–433, 2019.

[145] B. Nagel, D. Bohnke, V. Gollnick, P. Schmollgruber, J. Alonso, A. Rizzi, and G. La Rocca, "Communication in aircraft design: Can we establish a common language?," in *Proceedings of the 28th congress of the International Council of the Aeronautical Sciences, paper ICAS2012-1.9.1* (I. Grant, ed.), pp. 1–13, ICAS, 2012.

[146] KE-works, "Ke-chain, software package." [Online]. Available: https://ke-chain.com/platform/. Retrieved: 04 December 2024.

[147] Dassault Systèmes, "Isight & the simulia execution engine." https://www.3ds.com/products/simulia/isight. Accessed: 2024-12-06.

[148] V. Mohan, A. K. Jeyaraj, and S. Liscouët-Hanke, "Systems integration considerations for hybrid-electric commuter aircraft: Case study for the do-228," in *AIAA SCITECH 2023 Forum*, American Institute of Aeronautics and Astronautics, 2023. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2023-1361.

[149] S. Bonnet, J.-L. Voirin, D. Exertier, and V. Normand, "Not (strictly) relying on sysml for mbse: Language, tooling and development perspectives: The arcadia/capella rationale," in *Proceedings of the Annual IEEE Systems Conference (SysCon)*, (New York), pp. 1–6, Institute of Electrical and Electronics Engineers, 2016.

[150] N. Tabesh, A. K. Jeyaraj, S. Liscouët-Hanke, and A. Tamayo, "Integration of the functional hazard assessment within a model-based systems engineering framework," *Journal of Aerospace Information Systems*, vol. 0, no. 0, pp. 1–13, 2024. [Online]. Available: https://doi.org/10.2514/1.I011371.

[151] E. Clement, A. Rauzy, and T. Thomas, "Arbre analyste: Un outil d'arbres de défaillances respectant le standard open-psa et utilisant le moteur xfta," in *19e Congrès de Maîtrise des Risques et Sûreté de Fonctionnement*, (Dijon, France), Chaire Blériot-Fabre, Ecole Centrale de Paris, 2014.

[152] R. Sedgewick, *Algorithms in C, Part 5: Graph Algorithms, Third Edition*. Addison-Wesley Professional, third ed., 2001.

[153] A. Bourezg and H. Bentarzi, "Graph theory based reliability assessment software program for complex systems," in *Advances in Reliability Analysis and its Applications* (M. Ram and H. Pham, eds.), Springer Series in Reliability Engineering, Springer, Cham, 2020.

[154] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[155] A. B. Rauzy, *Probabilistic Safety Analysis with XFTA*. Trondheim, Norway: The AltaRica Association, 2020. [Online]. Available: https://altarica-association.org.

[156] E. Baalbergen, J. Vankan, L. Boggero, J. H. Bussemaker, T. Lefebvre, B. Beijer, A.-L. Bruggeman, and M. Mandorino, "Advancing cross-organizational collaboration in aircraft development," in *AIAA AVIATION 2022 Forum*, American Institute of Aeronautics and Astronautics, June 2022.

[157] DSD-DBS, "py-capellambse: A python library for working with capella models," 2024. Accessed: December 4, 2024.

[158] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd ed., 2009.

[159] I. van Gent, G. L. Rocca, and L. L. Veldhuis, *Composing MDAO symphonies: graph-based generation and manipulation of large multidisciplinary systems*. American Institute of Aeronautics and Astronautics, 2017.

[160] V. Mohan, "Framework and model development for aircraft systems integration in conceptual-level multidisciplinary design analysis," Master's thesis, Concordia University, December 2023. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/993415/.

[161] J. Bussemaker, L. Boggero, and P. D. Ciampa, "From system architecting to system design and optimization: A link between mbse and mdao," *INCOSE International Symposium*, vol. 32, no. 1, pp. 343–359, 2022.

[162] A. K. Jeyaraj, J. Bussemaker, S. Liscouet-Hanke, and L. Boggero, "Systems architecting: A practical example of design space modeling and safety-based filtering within the agile4.0 project," in *33rd Congress of the International Council of the Aeronautical Sciences, ICAS 2022*, September 2022.

[163] J. Blank and K. Deb, “pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[164] G. Licheva, V. Mohan, A. K. Jeyaraj, P. Bamrah, M. Mir, and S. Liscouet-Hanke, *System Integration Study for a Hybrid-Electric Commuter Aircraft Concept with a Solar Auxiliary Power System.* in AIAA SciTech 2024 Forum, American Institute of Aeronautics and Astronautics, 2024. doi:10.2514/6.2024-1538.

[165] G. Licheva, V. Mohan, P. Bamrah, N. Tabesh, H. Jahanara, M. Mir, and A. Jeyaraj, “Solar auxiliary power hybrid-electric aircraft,” in *IEEE/AIAA Electrified Aircraft Technologies Symposium, Student Design Challenge*, May 2023.

[166] K. R. Antcliff, M. D. Guynn, T. Marien, D. P. Wells, S. J. Schneider, and M. J. Tong, *Mission Analysis and Aircraft Sizing of a Hybrid-Electric Regional Aircraft.* American Institute of Aeronautics and Astronautics, January 2016. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2016-1028.

[167] K. R. Antcliff and F. M. Capristan, “Conceptual design of the parallel electric-gas architecture with synergistic utilization scheme (pegasus) concept,” in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. Accessed: Nov. 4, 2024.

[168] N. J. Blaesser, “Propeller-wing integration on the parallel electric-gas architecture with synergistic utilization scheme (pegasus) aircraft,” in *AIAA Scitech 2019 Forum*, 2019. Accessed: Nov. 4, 2024.

[169] F. M. Capristan and N. J. Blaesser, “Analysis of the parallel electric-gas architecture with synergistic utilization scheme (pegasus) concept,” Tech. Rep. NASA/TM–2019–220396, NASA Langley Research Center, 2019. [Online]. Available: https://ntrs.nasa.gov/citations/20190030874. Accessed: Nov. 4, 2024.

[170] I. Ordaz, E. J. Nielsen, and L. Wang, “Design of a distributed propulsion concept using an adjoint-based approach and blade element theory to minimize power,” in *AIAA AVIATION 2020 FORUM*, 2020. Accessed: Nov. 4, 2024.

[171] N. J. Blaesser and Z. J. Frederick, “Tail sizing considerations for wingtip propulsor driven aircraft applied to the parallel electric-gas architecture with synergistic utilization scheme (pegasus) concept,” in *AIAA AVIATION 2020 FORUM*, 2020. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2020-2633. Accessed: Nov. 4, 2024.

[172] N. J. Blaesser, Z. J. Frederick, I. Ordaz, F. Valdez, and S. Jones, “Mission and vehicle-level updates for the parallel electric-gas architecture with synergistic utilization scheme (pegasus) concept aircraft,” tech. rep., NASA Technical Memorandum NASA/TM-20240001480, Langley Research Center, Hampton, Virginia, March 2024. [Online]. Available: https://ntrs.nasa.gov/citations/20240001480.

[173] NASA Systems Analysis and Concepts Directorate, Langley Research Center, “Pegasus.” https://sacd.larc.nasa.gov/asab/asab-projects-2/pegasus/, May 2024. Accessed: Nov. 4, 2024.

[174] Bombardier, “Mighty wings,” 2024. Available: https://bombardier.com/en/experience/featuredaircraft/mighty-wings. [Accessed: Oct. 8, 2024].

[175] C. D. Rodriguez, "An architecture-based weight estimation method for aircraft fuel systems," Master's thesis, Concordia University, August 2022. Available: https://spectrum.library.concordia.ca/id/eprint/991221/.

[176] U.S. Federal Aviation Administration, "Fuel pumps." 14 CFR § 25.991, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 25, Subpart E. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-E/subject-group-ECFR30ebd8ef16ae85f/section-25.991.

[177] U.S. Federal Aviation Administration, "Fuel pumps." 14 CFR § 23.991, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 23. Available: https://www.govinfo.gov/content/pkg/CFR-2012-title14-vol1/pdf/CFR-2012-title14-vol1-sec23-991.pdf.

[178] N. Kuelper, A. K. Jeyaraj, S. Liscouët-Hanke, and F. Thielecke, "Integration of a model-based systems engineering framework with safety assessment for early design phases: A case study for hydrogen-based aircraft fuel system architecting." Manuscript under review.

[179] U.S. Federal Aviation Administration, "Designation of Applicable Regulations, 14 CFR § 21.17." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-21/subpart-B/section-21.17. Accessed: 2024-12-08.

[180] U.S. Federal Aviation Administration, "Airworthiness Standards: Normal Category Rotorcraft, 14 CFR Part 27." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-27?toc=1. Accessed: 2024-12-08.

[181] U.S. Federal Aviation Administration, "Airworthiness Standards: Transport Category Rotorcraft, 14 CFR Part 29." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-29. Accessed: 2024-12-08.

[182] U.S. Federal Aviation Administration, "Airworthiness Standards: Manned Free Balloons, 14 CFR Part 31." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-31?toc=1. Accessed: 2024-12-08.

[183] U.S. Federal Aviation Administration, "Airworthiness Standards: Propellers, 14 CFR Part 35." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-35. Accessed: 2024-12-08.

[184] U.S. Federal Aviation Administration, "Airworthiness Criteria: Special Class Airworthiness Criteria for the Joby Aero, Inc. Model JAS4–1 Powered-Lift." [Online]. Available: https://www.federalregister.gov/documents/2022/11/08/2022-23962/airworthiness-criteria-special-class-airworthiness-criteria-for-the-joby-aero-inc-model-jas4-1. Accessed: 2024-12-08.

[185] U.S. Federal Aviation Administration, "System Power Generation, Storage, and Distribution, 14 CFR § 23.2525." https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23/subpart-F/section-23.2525. Accessed: 2024-12-08.

[186] ASTM International, "Standard Practice for Safety Assessment of Systems and Equipment in Small Aircraft," 2024. [Online]. Available: https://www.astm.org/f3230-24.html. Accessed: 2024-12-08.

[187] J. Bevirt, A. Stoll, M. van der Geest, S. MacAfee, and J. Ryan, “Fault-tolerant vtol aircraft with redundant control surfaces and actuators,” Sep. 2024. Available: https://patents.google.com/patent/US20240300660A1/en?assignee=Joby+Aviation&oq=Joby+Aviation&sort=new.

[188] V. Gadekar, “Joby s4 evtol.” VSP Airshow, 2024. Available: https://airshow.openvsp.org/vsp/LtyRrSrYXRhQlDkYwPDX, Accessed: Nov. 7, 2024.

[189] U.S. Federal Aviation Administration, “Design Considerations for Minimizing Hazards Caused by Uncontained Turbine Engine and Auxiliary Power Unit Rotor Failure.” Advisory Circular AC 20-128A, Mar. 1997. Available: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentid/22187.

[190] U.S. Federal Aviation Administration, “Installation.” 14 CFR § 25.901, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 25, Subpart E. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-E/subject-group-ECFR3db216ad9d52259/section-25.901.

[191] U.S. Federal Aviation Administration, “Engines.” 14 CFR § 25.903, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 25, Subpart E. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-E/subject-group-ECFR3db216ad9d52259/section-25.903.

[192] R. H. Jansen, C. C. Kiris, T. Chau, G. K. W. Kenway, L. G. Machado, J. C. Duensing, A. Mirhashemi, J. M. Haglage, T. P. Dever, J. W. Chapman, B. D. French, T. W. Goodnight, L. R. Miller, J. S. Litt, C. L. Denham, M. Lynde, R. Campbell, B. Hiller, and N. Heersema, “SUSAN Transport Aircraft Concept and Trade Space Exploration,” in *AIAA SciTech Forum and Exposition*, American Institute of Aeronautics and Astronautics, Jan. 2022.

[193] J. Haglage, T. Dever, R. Jansen, and M. Lewis, “Electrical system trade study for susan electrofan concept vehicle,” in *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, 2022. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2022-2183. PDF: https://arc.aiaa.org/doi/pdf/10.2514/6.2022-2183.

[194] NASA Glenn Research Center, “SUSAN Animation.” https://www.grc.nasa.gov/WWW/GVIS/SUSAN/SUSANembed.html. Accessed: May 2024.

[195] J. S. Litt, J. L. Kratz, S. Bianco, J. Sachs-Wetstone, T. Dever, H. E. Buescher, N. C. Ogden, F. Valdez, D. W. Budolak, M. J. Boucher, A. P. Patterson, and R. Jansen, “Control architecture for a concept aircraft with a series/parallel partial hybrid powertrain and distributed electric propulsion,” in *AIAA SCITECH 2023 Forum*, AIAA 2023-1750, American Institute of Aeronautics and Astronautics, Jan. 2023.

[196] P. Bamrah, “Zonal safety and particular risk analysis for early aircraft design using parametric geometric modelling,” Master’s thesis, Concordia University, July 2023. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/992615/.

[197] P. Bamrah, S. Liscouet-Hanke, A. Tfaily, and A. Tamayo, "Zonal safety and particular risk analysis for aircraft conceptual design," in *AIAA AVIATION 2023 Forum*, American Institute of Aeronautics & Astronautics, 2023. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2023-4197.

[198] F. Sanchez and S. Liscouët-Hanke, "Thermal risk prediction methodology for conceptual design of aircraft equipment bays," *Aerospace Science and Technology*, vol. 104, p. 105946, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963820306283.

[199] U.S. Federal Aviation Administration, "Designated fire zones; regions included." 14 CFR § 25.1181, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 25, Subpart E. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-E/subject-group-ECFRee0f409daee8550/section-25.1181.

[200] U.S. Federal Aviation Administration, "Flammable fluid fire protection." 14 CFR § 25.863, Code of Federal Regulations, Title 14, Chapter I, Subchapter C, Part 25, Subpart D. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-25/subpart-D/subject-group-ECFR1e1f52030ba4797/section-25.863.

[201] Bombardier Inc., *BD500 Series (PW PW1500G) Initial Maintenance Training Course Technical Training Manual.* Bombardier Customer Training, Montreal, Quebec, Canada, 2024. Module 1: General Familiarization, page 27-12.

[202] Airbus, *Airbus A330 Flight Crew Operating Manual (FCOM) - Hydraulics.* [Online]. Available: https://www.smartcockpit.com/docs/A330-Hydraulic.pdf. Accessed: 2024-10-15.

[203] Airbus, *Airbus A340 Flight Crew Operating Manual (FCOM) - Hydraulics.* [Online]. Available: https://www.smartcockpit.com/my-aircraft/airbus-a340/#. Accessed: 2024-10-15.

[204] Airbus S.A.S, *A350-900 Flight Deck and Systems Briefing for Pilots*, 2011. Issue 02 - Sept 2011.

[205] D. van den Bossche and Airbus, "The a 380 flight control electrohydrostatic actuators, achievements and lessons learnt," in *Proceedings of the 25th International Congress of the Aeronautical Sciences*, 2006.

[206] "737 rudder system." http://www.b737.org.uk/theruddersystem.htm. [Accessed: Dec. 5, 2024].

[207] U.S. Federal Aviation Administration, "Rudder system scenarios." https://www.faa.gov/sites/faa.gov/files/2022-11/Boeing%20rudder%20system%20scenarios.pdf. [Accessed: Dec. 5, 2024].

[208] Y. C. Yeh, "Triple-triple redundant 777 primary flight computer," in *1996 IEEE Aerospace Applications Conference. Proceedings*, vol. 1, pp. 293–307, IEEE, 1996.

[209] Aviation Learning, "Boeing 777 refresher course." Technical Training Manual. June 2004, p. 156.

[210] C. Wang, I.-S. Fan, and S. King, “Failures mapping for aircraft electrical actuation system health management,” $PHME_CONF, vol. 7, pp. 509 - -520, June$2022.

[211] Boeing Commercial Airplane Company, “Boeing 747 primary flight control systems reliability and maintenance,” nasa contractor report 159010, NASA, Apr. 1979. Aviation Energy Efficiency Program. [Online]. Available: https://ntrs.nasa.gov/api/citations/19810018578/downloads/19810018578.pdf.

[212] Commercial Aircraft Corporation of China (COMAC), *C919 Aircraft Test Flight Maintenance / Crew Textbook: Main Flight Control System*, 2024. For educational use only. https://max.book118.com/html/2019/0711/6012223132002044.shtm.

[213] SJI Training Center, *ATA 27 Flight Controls, Technical Training Manual.* Sukhoi Civil Aircraft, 2011. Page 7.

[214] Dassault Aviation, *Falcon 7X Flight Controls*, 2018. Page 9. [Online]. Available: https://www.smartcockpit.com/my-aircraft/dassault-falcon-7x/. Accessed: Oct. 4, 2024.

[215] Ivan Luciani, “G650 Flight Control System.” Online. Available: https://www.code450.com/s/g650_flight_control_system-4dlk.pdf, 2018. For study purposes only, unofficial compilation.

[216] Embraer, *Embraer 190 Flight Controls*, 2017. Page 17. [Online]. Available: https://www.smartcockpit.com/my-aircraft/embraer-erj-190-195/. Accessed: Oct. 4, 2024.

[217] K. S. Bohra and Y. S. Dharmadhikari, “History of fly by wire, the tech that expanded human horizons in aviation,” in *2023 8th IEEE History of Electrotechnology Conference (HISTELCON)*, pp. 46–51, 2023.

[218] Y. C. Yeh, “Airbus fly-by-wire computers a320 – a350,” in *IEEE ComSoc Technical Committee on Communication Quality & Reliability, Emerging Technology Reliability Roundtable*, (Stevenson, WA, USA), May 2016. [Online]. Available: https://cqr2016.ieee-cqr.org/06-ETR-RT16_Yeh.pdf. Accessed: Nov. 7, 2024.

[219] D. Briere and P. Traverse, “Airbus a320/a330/a340 electrical flight controls - a family of fault-tolerant systems,” in *FTCS-23 The Twenty-Third International Symposium on Fault-Tolerant Computing*, pp. 616–623, 1993.

[220] F. George, “Pilot report: Bombardier global 7500,” 2020. [Online]. Available: https://aviationweek.com/business-aviation/aircraft-propulsion/pilot-report-bombardier-global-7500. Accessed: 2025-01-09.

[221] H. A. Rediess, “Foreign technology summary of flight crucial flight control systems,” in *NASA Aircraft Controls Research Proceedings, 1983*, (Langley Research Center, Hampton, VA, USA), March 1984. Document ID: 19840012519, Accession Number: 84N20587, CONTRACT_GRANT: NAS1-17403, Work of the US Gov. Public Use Permitted. [Online]. Available: https://ntrs.nasa.gov/citations/19840012519.

[222] J. Gautrey, “Flight control system architecture analysis and design for a fly-by-wire generic regional aircraft,” Tech. Rep. Report No. 9604, College of Aeronautics, Cranfield University, Cranfield, Bedford, England, March 1996. Flight Dynamics Group. [Online]. Available: https://dspace.lib.cranfield.ac.uk/items/63326ce3-5e68-4a7d-987e-62fbc7650e89.

[223] Matt Thurber, "Pilot Report: Bombardier Global 7500." Online. Available: https://aviationweek.com/business-aviation/aircraft-propulsion/pilot-report-bombardier-global-7500, 2018. Accessed: Oct. 4, 2024, page 8.

[224] Moog Inc., *Moog Boeing 787 Datasheet*, 2017. Page 2. [Online]. Available: https://www.moog.com/content/dam/moog/literature/Aircraft/commercial_aftermarket/moog-boeing787-datasheet.pdf. Accessed: Oct. 4, 2024.

[225] Airbus S.A.S., *A350-900 Flight Deck and Systems Briefing for Pilots, Issue 02*. Airbus S.A.S., 2011. Page 27.5.

[226] AOPA - Aircraft Owners and Pilots Association, "A Fly-by-Wire Future." Online. Available: https://www.aopa.org/news-and-media/all-news/2020/october/pilot/a-fly-by-wire-future, 2020. Accessed: Oct. 4, 2024.

[227] Aviation Pros, "Thales New Rudder-by-Wire Flight Control System Will Fly on Cessna Citation Longitude." [Online]. Available:https://www.aviationpros.com/tools-equipment/insurance-finance/aircraft/press-release/12140535/thales-thales-new-rudder-by-wire-flight-control-system-will-fly-on-cessna-citation-longitude, 2015. Accessed: Oct. 4, 2024.

[228] J.-C. Maré, *Aerospace Actuators 1, Reliability*, ch. 2, pp. 33–61. John Wiley Sons, Ltd, 2016.

[229] R. DeLucia, J. Salvino, and B. Fenton, "Statistics on aircraft gas turbine engine rotor failures that occurred in u.s. commercial aviation during 1984," Tech. Rep. DOT/FAA/CT-89/6, FAA Technical Center, Naval Air Propulsion Center, June 1989. Available through the National Technical Information Service.

[230] Bombardier, "Special check/modification - emergency ac-power supply - replacement of the ram air turbine (rat) deployment actuator part no. bz02001-01 (gl456-1301-1)," tech. rep., Bombardier Aerospace, 2019. Basic Issue: Feb 22, 2019. Available under the title Service Bulletin 700-24-5015. [Online]. Available: https://downloads.regulations.gov/FAA-2020-0104-0005/attachment_3.pdf.

[231] Bombardier Inc., "Bombardier marks 10th anniversary of the first global 5000 delivery," 2015. [Online]. Available: https://bombardier.com/en/media/news/bombardier-marks-10th-anniversary-first-global-5000-delivery. Accessed: 2024-12-08.

[232] N. Williard, W. He, C. Hendricks, and M. Pecht, "Lessons learned from the 787 dreamliner issue on lithium-ion battery reliability," *Energies*, vol. 6, no. 9, pp. 4682–4695, 2013.

[233] RSL Electronics Ltd., "Advanced brake controller (abc)." [Online]. Available: https://www.rsl-electronics.com/RSL/Templates/showpage.asp?DBID=1&LNGID=1&TMID=108&FID=1328&IID=1770. Accessed: 2024-12-08.

[234] I. Moir, A. Seabridge, and M. Jukes, *Safety Analysis – Electrical System*, pp. 543–545. John Wiley Sons, Ltd, 2013.

[235] R. G. Arno, "Nonelectronic parts reliability data - 2 (nprd-2)," Tech. Rep. ADA108387, Reliability Analysis Center, Rome Air Development Center, Summer 1981. Prepared under contract to the Rome Air Development Center.

[236] E. Kolip, "Improving model-based system architecture specification to enable fault tree analysis," Master's thesis, Concordia University, May 2024. https://spectrum.library.concordia.ca/id/eprint/994053/.

Appendix A

Rule-Based Safety Assessment

A.1 Analysis of system architecture implementation (Bottom-up)

Table A.1 shows a summary of the rudder actuation architecture for typical large commercial aircraft and some business aircraft. Most aircraft feature hydraulic actuation of the rudder except for the Gulfstream G650, Airbus A350 and Airbus A380, which feature a combination of hydraulic actuators with electrically powered actuators (EHA) or hydraulic actuators with an electric backup system (EBHA). The aircraft with purely hydraulic actuation feature three independent hydraulic systems, and those with partial electrically powered actuators feature a two hydraulic and two electrical system architecture. Typically, three actuators are allocated to the rudder, with some exceptions.

Table A.1: Hydraulic systems and actuator technologies of various aircraft

Aircraft	**No. of Hydraulic Systems**	**Power Transfer Unit**	**No. of Surfaces**	**No. Actuators per Surface**	**Actuator Technology**
Airbus A220 [201]	3	Yes	1	3	EHSA
Airbus A320 [201, p. 3]	3	Yes	1	3	EHSA
Airbus A330 [202, p. 4]	3	No	1	3	EHSA
Airbus A340 [203, p. 4]	3	No	1	3	EHSA
Airbus A350 [204, p. 193]	2	No	1	3	EHSA, EHSA, EHA
Airbus A380 [205, p. 2]	2	No	2	4	EHSA, EBHA
Boeing 737 [206, 207]	2	Yes	1	2	EHSA, EHSA
Boeing 777 [208, 209]	3	No	1	3	EHSA
Boeing 787 [210]	3	No	1	3	EHSA
Boeing 747 [211, pp. 31–37]	4	Yes	2	1	EHSA
Comac C919 [212]	3	-	1	3	EHSA
Sukhoi Superjet [213]	3	Yes	1	3	EHSA
Dassault Falcon 7X [214, p. 14]	3	No	1	1	EHSA
Gulfstream G650 [215]	2	Yes	1	2	EHSA, EBHA
Embraer 190 [216]	3	Yes	1	2	EHSA

The Airbus A380 has a split rudder with two surfaces featuring two actuators each, while the Boeing 737 has one rudder surface with two actuators, one being a dual actuator and the other a backup actuator [206, 207]. The Boeing 747 also has a split rudder surface with one actuator per surface, and in this case, the actuator is a dual actuator unit that can take hydraulic power supply from two different hydraulic systems [211, p. 35]. Similarly, the Falcon 7X rudder actuator is also a dual actuator [214, p. 14].

The Gulfstream G650 features the use of an EBHA alongside an EHSA for rudder actuation. It has two hydraulic systems and two main electrical systems with one backup electrical system. The EBHAs are supplied by the backup bus, which is linked to the left main electrical bus as well. Under normal operation, the two main buses and the emergency bus are supplied with normal power from the engines. Upon the loss of the left main generator, power from the right main generator can be routed through to the left electrical bus using tie contactors. Two independent electrical sources, i.e., RAT and an

EBHA battery, are linked to the emergency bus to power the EBHAs in case all other power sources are unavailable.

A.2 Analysis of existing FbW architectures

The earliest FbW implementation on commercial aircraft was the analog FbW system on the Concorde [217], which also featured a complete mechanical backup. Other implementations feature full quadruplex FCC architectures on the F114 Nighthawk and other aircraft. Airbus introduced the first digital FbW system aboard the Airbus A320, and subsequent aircraft such as the A340, A350, and A380 also feature FbW systems. Boeing introduced its first FbW system aboard the Boeing 777 aircraft. Initially, alongside the FbW system, two spoilers and the trimmable horizontal stabilizer were signalled mechanically, but these were also made to be electrically controlled in later versions of the aircraft.

As with any safety-critical system aboard an aircraft, the FbW system architecture is specified, built, and installed according to the principles of independence, dissimilarity, and segregation. Airbus initially had two elevator and aileron (ELAC) computers and three spoiler and elevator (SEC) computers. The ELACs and the SECs had two lanes each in a dual-dual architecture and triple-dual architecture, respectively [218]. The ELACs and the SECs are based on dissimilar hardware made by different manufacturers (Thomson-CSF and SFENA/Aerospatiale) [219, p. 619]. Additionally, four different software packages are implemented to mitigate against software-related failures [219, p. 620].

System segregation practices are applied by placing the FCCs in three different locations and routing the links along different paths such as overhead, underfloor, and through the cargo compartment. Despite these measures, an ultimate backup in the form of a mechanical link to the THS and the rudder is maintained in the unlikely event of the failure of all FCCs. However, the FbW system architecture is shown to meet safety requirements without considering the backup mechanical system.

A failure of ELAC2 results in ELAC 1 taking over, and subsequent failure of ELAC 1 results in SECs taking over control. SEC3 is used to control inboard spoilers. Briere et al. note that safety objectives can be met with three computers, with the additional computer (excluding SEC3) satisfying operational constraints, i.e., being able to take off and land with one computer failed or not operational [219, p. 619].

In the A340-600, Airbus introduced three primary flight control computers and three secondary flight control computers. Additionally, the A340-600 has an electrical backup system (EBS) implemented by a backup control module (BCM) with its own backup power supply (BPS) [218, p. 6]. The BCM signals two of the three rudder actuators and replaces the mechanical link to the rudder. A similar arrangement is also seen on the A350 and A380 aircraft. On the A350 the EBS controls the board ailerons, elevator and rudder in a situation when all PRIMs and SECs have failed. The EBS consists of a BPS, which has a generator that is supplied by the yellow hydraulic system [218, p. 10]. The BCM is another example of introducing dissimilarity in the FbW system, as the BCM signals to the actuators are analog when compared to the digital FCC signals. The BCM is also used in the Airbus A380, and a similar approach of introducing dissimilarity using analog signalling is also implemented in the Airbus A220 using the Alternate Flight Control Unit. Similar backup systems are also used on other aircraft, such as the Bombardier Global 7500 [220, p. 8]. The analog backup control system has replaced the need to have a mechanical backup system for primary flight controls on modern aircraft. This design philosophy mirrors those adopted in the development of digital FbW systems for fighter aircraft [221, p. 373].

Boeing's implementation of FbW on the 777 uses three FCCs with three lanes each. The aircraft can be dispatched with up to one lane inoperational on a single FCC and can be operated for a short time with one FCC inoperational. The FbW system on the 777 uses a distributed approach with the central FCCs signalling actuator control electronics which in turn convert the digital FCC signals to analog actuator commands. The 777 has four ACEs that are allocated as two per control surface except the rudder, to which three ACEs are allocated. In normal law, the ACEs, pilot inceptors and air data sensors provide information to the FCC, which then generates control commands and sends them back to the ACEs, which then control the actuators accordingly. In direct law, the actuators are directly signalled from the pilot command inceptors through the ACEs without any FCC commands being issued, as under direct law, it is assumed that the FCCs have failed or are non-operational. Consider figure A.1, which is a recreation of the power and control architecture of the 777 FbW system focusing specifically on the ACEs using the generic element descriptor.

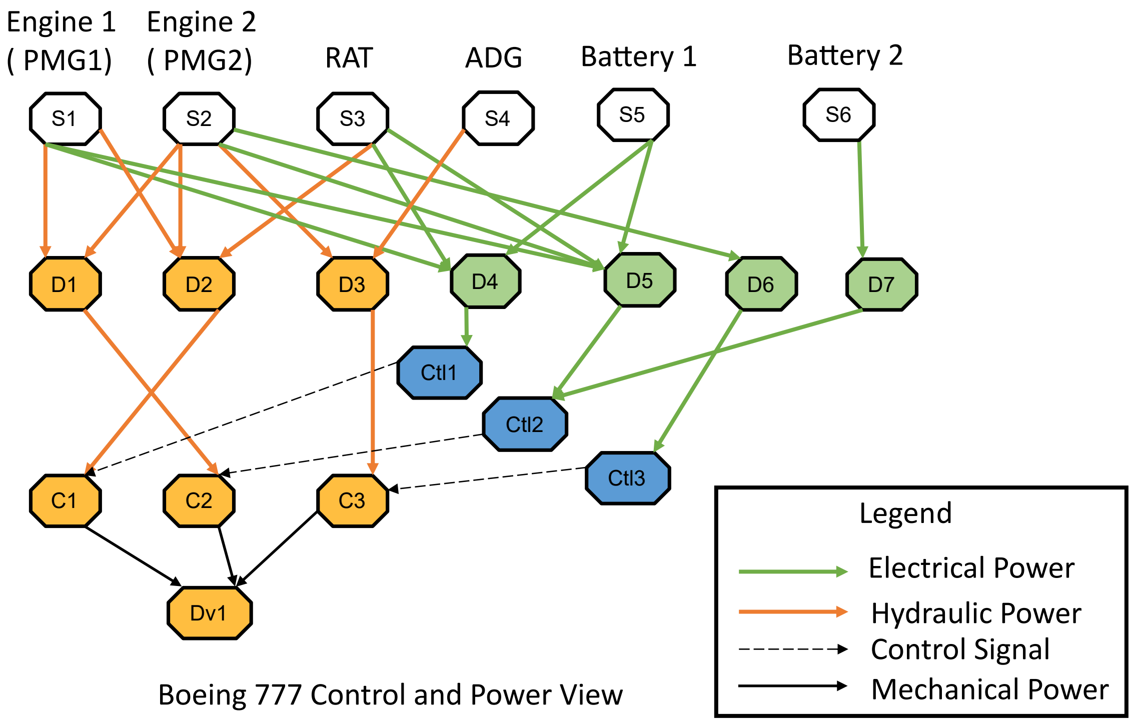


Figure A.1: Power and Control View of Boeing 777 FbW - ACE Architecture

Here, it is evident that each rudder actuator is assigned a control element (in this case, an ACE) and that at least one actuator receives power from two sources. Furthermore, the sources of power for Ctl2 through D5 consist of both primary power sources, such as the engines (through PMGs), time-limited emergency sources, such as dedicated flight control batteries, and non-time-limited sources, such as the RAT. Thus in the case of all engine failure, the rudder can still be controlled by at least one controller drawing power from a variety of emergency power sources. Another notable feature of the 777 is that the power supply for the FbW system is completely independent of the electrical supply to other electrical consumer systems. It is also noteworthy that an explicit dissimilar path or ultimate backup is not immediately apparent from a top-level description of the FbW control architecture. However, in [222, p. 121], Gautrey describes the ACEs receiving a direct analog link from the inceptor sensors, which is supposedly a completely independent part of the ACE. This suggests that a failure of the ACE may not necessarily imply the failure of the analog channel within the ACE. This particular feature adds a dissimilar control path to the FbW control architecture and is comparable to the ultimate backup analog control in other FbW aircraft, such as the A350, A380, and A220.

Modern FbW aircraft, such as the A220, feature a distributed actuator control using REUs. These units can be positioned in unpressurized compartments closer to the actuator they control. REUs can be single- or dual-channelled and can control just one actuator or simultaneously control two actuators, respectively. In direct mode, the FbW system relies on sending signals from the inceptor sensors directly to the REU, and therefore, the allocation of REUs to actuators and surfaces is important from a safety perspective. Table A.2 provides a summary of the controller configurations of a variety of modern FbW aircraft.

Table A.2: Overview of typical actuator electronics and FCC configuration for FbW aircraft

Aircraft	FbW type	Actuator controller type	No. of controllers	No. of primary flight control surfaces	Category	No. PFCC	No. SFCC	No. Dissimilar Backup Computers
A220 [201]	Full FbW	REU	10	15	Distributed	3	-	1
G650 [215]	Full FbW	REU & MCE	9 & 7	11	Distributed	2	-	1
Global 7500 [223]		REU	11	13	Distributed	3		1
Falcon 7X [214]		ACE	4	11	Distributed	3	3	1
Boeing 777 [208, 222]	Full FbW	ACE	4	-	Distributed	3	-	-
Boeing 787 [224]	Full FbW	REU & MCE	25 & 6	17	Distributed	3	-	-
Embraer 190 [216]	Partial Analog FbW	ACE	3	12	Centralized	4	-	-
Sukhoi Superjet [213]	Full FbW	ACE	14	15	Distributed	3	-	-
Comac C919 [212]	Full FbW	ACE & REU	4 & 19	15	Distributed	3	-	-
Airbus A350 [225]	Full FbW	-	-	21	Centralized	3	3	1
Embraer Praetor 600 [226]	Full FbW	REU	26	13	Distributed	2	-	1
Airbus A380	Full FbW	-	28	-	Centralized	3	3	1
Cessna Citation Longitude [227]	Partial FbW (Rudder)	SECU	2	1 FbW Controlled Surface	-	-	-	-

Legend:

- FbW: Fly-by-Wire
- REU: Remote Electronics Unit
- MCE: Motor Control Electronics
- ACE: Actuator Control Electronics
- PFCC: Primary Flight Control Computer
- SFCC: Secondary Flight Control Computer
- SECC: Smart Electronic Control Unit

In the context of this analysis, the REUs, ACEs, MCUs, and SECUs can be treated differently according to the implementation on a specific aircraft. For example, the functions of the ACE and REU are similar in that they both close a control loop with the actuator and convert digital commands into analog input for the actuator. However, on the Boeing 777, the ACEs play a more central, more centralized role, but on the Boeing 787, the REUs perform a similar function but are distributed across different surfaces and actuators. In addition to functional differences between ACEs and REU, there can also be terminological differences from one implementation to another. For example, on the Comac C919 FbW architecture, the ACEs are centralized, and the term REU is used to refer to an electronic unit locally at the actuator that performs the digital-to-analog conversion. However, on the Airbus A220, the REUs directly interface with the FCC and the pilot command interface (IIM) and provide command signals to the actuator. Another example is the Sukhoi Superjet, where the ACEs are distributed in a way that is akin to REUs on the A220. On the Superjet, two types of ACEs are used, which enables the FbW system to benefit from added dissimilarity.

In addition to the ACEs and REUs, which control the actuator servo-valve when interfaced with hydraulic actuators, an MCE controls the electric motor driving the hydraulic pump on EHAs and EBHAs, and as a result, aircraft featuring EHAs and EBHAs such as the G650 and the Boeing 787s have a combination of both REUs and MCEs distributed across different actuators and control surfaces.

Control Allocation for FbW aircraft

This section outlines a notional architecture of Remote Electronic Units (REUs) assigned to primary flight control actuators for the Airbus A220 aircraft based on the rules developed in 3.3.4 and limited publicly available information about the aircraft.

The Airbus A220 (formerly known as CSeries - CS100, CS300) is a tube-wing aircraft with two sets of ailerons, two elevators, a single rudder, four multi-functional spoilers, and two ground spoilers. The aircraft has three hydraulic systems powering hydraulic actuators, which are controlled by a full FbW system. The FbW system relies on three flight control primary computers and a set of REUs that are distributed across the aircraft and typically positioned close to the actuators themselves. Two or more REUs are assigned to each surface, and the REUs are thereby assumed to be dual-channelled, counting towards a total of 10 REUs. A notional allocation of the REUs to control the actuators on each surface is shown in figure A.2.

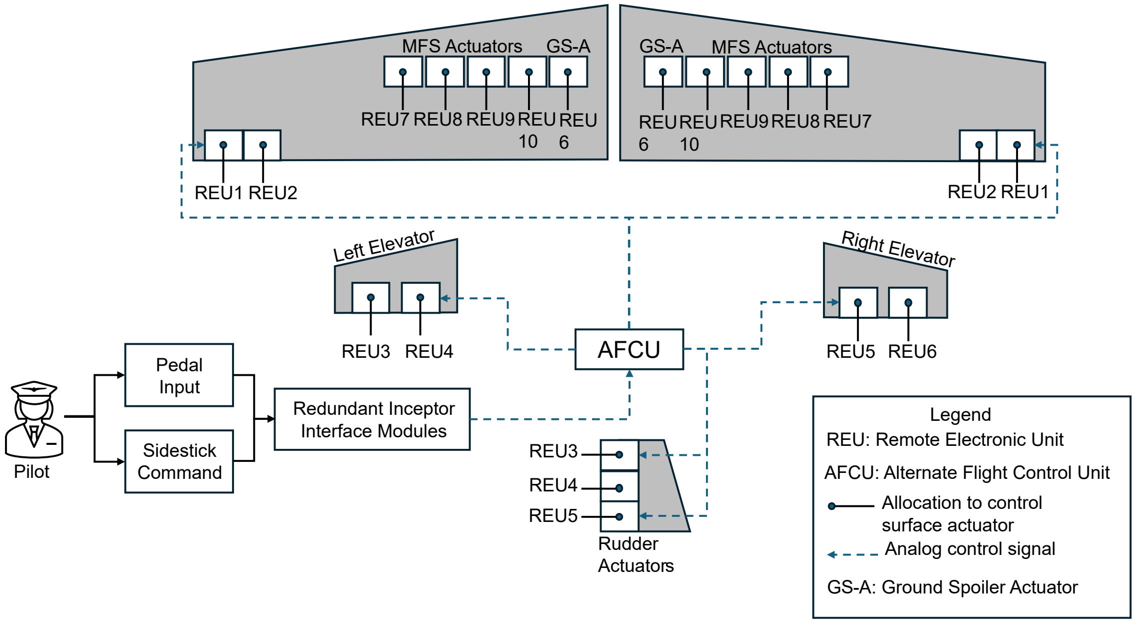

MFS Actuators
GS-A
REU7 REU8 REU9 REU 10 REU 6
REU1 REU2
GS-A
MFS Actuators
REU 6 REU 10 REU9 REU8 REU7
REU2 REU1
Left Elevator
REU3 REU4
Right Elevator
REU5 REU6
AFCU
Pedal Input
Sidestick Command
Pilot
Redundant Inceptor Interface Modules
REU3
REU4
REU5
Rudder Actuators
Legend
REU: Remote Electronic Unit
AFCU: Alternate Flight Control Unit
Allocation to control surface actuator
Analog control signal
GS-A: Ground Spoiler Actuator

Figure A.2: Notional REU allocation to primary control surface actuators for the Airbus A220

A.3 Summary of Safety Rules

This section compiles the safety rules discussed in sections 3.3.3 and 3.3.5.

A.3.1 Hydraulically Powered Rudder-based Yaw Control Actuation

Table A.3: Summary of safety rules pertaining to rudder-based yaw control systems actuated with hydraulic power

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 2, Rule 3
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 3

A.3.2 Electrically Powered Rudder-based Yaw Control Actuation

Table A.4: Summary of safety rules pertaining to rudder-based yaw control systems actuated with electrical power

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1, Rule 3, Rule 5
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 2, Rule 3, Rule 4
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 3, Rule 6

A.3.3 Mixed Technology Rudder-based Yaw Control Actuation

Table A.5: Summary of safety rules pertaining to rudder-based yaw control systems featuring mixed actuation technologies (electrical and hydraulic)

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 1, Rule 2
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 1, Rule 2

A.3.4 Flight Control Computer Allocation for FbW Yaw Control Actuation

Table A.6: Summary of safety for Flight Control Computer allocation on rudder-based FbW yaw control systems

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1, Rule 2, Rule 3, Rule 5, Rule 6
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 4, Rule 5, Rule 8
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 4, Rule 7

A.3.5 Remote Electronics Unit Allocation for FbW Yaw Control Actuation

Table A.7: Summary of safety rules for Remote Electronics Unit allocation pertaining to rudder-based FbW yaw control systems

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 2,
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 3, Rule 5, Rule 6
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 4

A.3.6 Yaw Control Actuation on Novel Aircraft - Multiple Rudders or Unconventional Aircraft Configurations

Table A.8: Summary of safety rules for yaw control on unconventional aircraft configurations

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 1,Rule 2, Rule 3

A.3.7 Flight Control Computer Allocation for Yaw Control on Unconventional Aircraft Configurations

Table A.9: Flight Control Computer allocation for FbW yaw control on unconventional aircraft configurations

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1, Rule 2, Rule 3
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 2
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 4

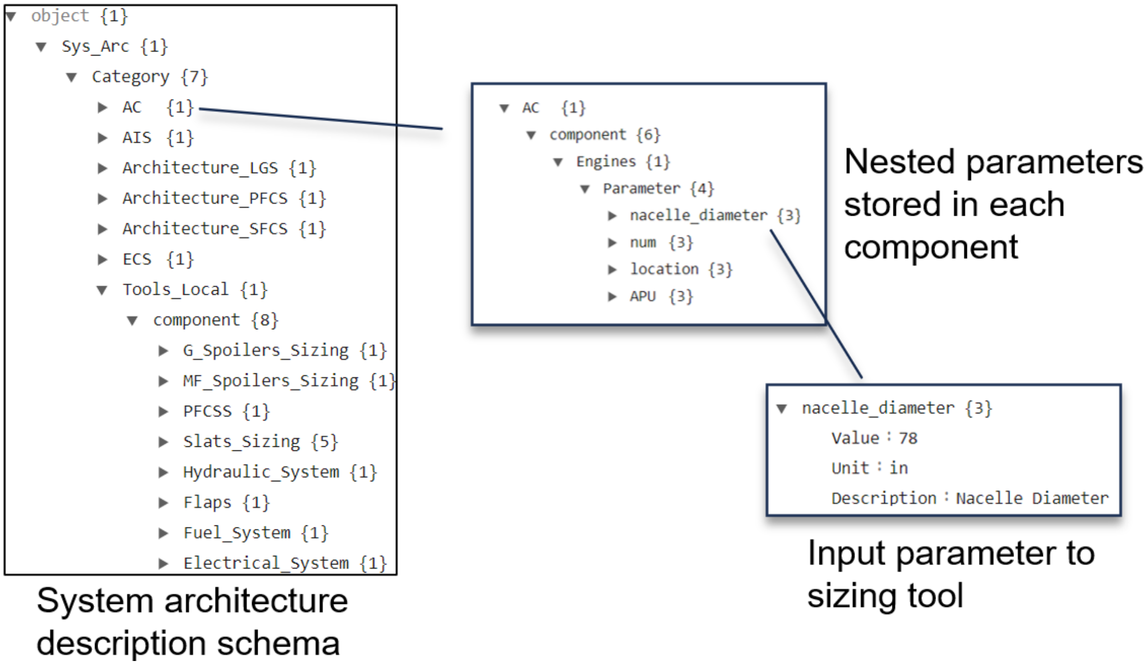
A.3.8 Remote Electronics Unit Allocation for Yaw Control on Unconventional Aircraft Configurations

Table A.10: Remote Electronics Unit allocation for FbW yaw control on unconventional aircraft configurations

Certification Rule	Category	Includes	Allocated Heuristic
14 CFR Part 25			
Subpart B 25.147	Directional and lateral control	25.147 (subparts a-1,b-1)	Rule 1, Rule 2
Subpart F 25.1310	Power source capacity and distribution	25.1310 (subpart a 1-4)	Rule 3, Rule 4
Subpart D 25.671	Control Systems	25.671 (subparts c 1-3, d)	Rule 3, Rule 4

A.4 Graph descriptor to MDAO link

System architecture description schema for industrial workflow



```
▼ object {1}
  ▼ Sys_Arc {1}
    ▼ Category {7}
      ▶ AC  {1}
      ▶ AIS {1}
      ▶ Architecture_LGS {1}
      ▶ Architecture_PFCS {1}
      ▶ Architecture_SFCS {1}
      ▶ ECS {1}
      ▼ Tools_Local {1}
        ▼ component {8}
          ▶ G_Spoilers_Sizing {1}
          ▶ MF_Spoilers_Sizing {1}
          ▶ PFCSS {1}
          ▶ Slats_Sizing {5}
          ▶ Hydraulic_System {1}
          ▶ Flaps {1}
          ▶ Fuel_System {1}
          ▶ Electrical_System {1}
```

System architecture description schema

```
▼ AC  {1}
  ▼ component {6}
    ▼ Engines {1}
      ▼ Parameter {4}
        ▶ nacelle_diameter {3}
        ▶ num {3}
        ▶ location {3}
        ▶ APU {3}
```

Nested parameters stored in each component

```
▼ nacelle_diameter {3}
    Value : 78
    Unit : in
    Description : Nacelle Diameter
```

Input parameter to sizing tool

Figure A.3: System architecture description schema for system sizing tool input

Appendix B

ASSESS L1-M1

B.1 Failure rate data

Table B.1 shows the failure rate data used as input for the case studies in chapter 4.1.

Table B.1: Failure rate data for aircraft power system, braking system and flight control actuation system architecture components

Component	Failure Rate per FH	Stance	Sources
Hydraulic Distribution	10^{-4}	Conservative	[228, p. 51]
Electrical Distribution	5×10^{-5}	Conservative	Assumption
EHSA	1×10^{-6}	-	[228, p. 51]
EBHA	5×10^{-6}	Conservative	Assumption
EHA	1×10^{-5}	-	[228, p. 51]
Turbine Engine	2.2×10^{-6}	Conservative	[229, p. 12]
Ram Air Turbine	10^{-4} (assumption) 2.661×10^{-6}	Mixed	[230, p. 2] [231]
Battery (Aircraft Li-Ion)	4×10^{-5}	Conservative	[232, p. 4689]
Brake Control Unit	6.61×10^{-5}	-	[233]

The failure rate data has been compiled from a variety of publicly available literature. In cases where the reliability of the data is difficult to ascertain, certain conservative assumptions are used instead. Take, for instance, the case of the RAT. Moir et al. use a failure rate of 10^{-3} failures per FH in an example demonstrating fault tree analysis for an aircraft electrical system in [234]. Their fault tree associates this failure rate with the "Loss of emergency AC". However, it is unclear whether this represents a failure mode where the generator associated with the RAT fails, whether the RAT fails to deploy, whether the RAT's blades fail, whether there is a mechanical failure in the RAT's gearbox or any other potential failure that could cause the complete loss of emergency power from the RAT. Therefore, to deal with the uncertainty in the failure rate, this thesis adopts a conservative value and an alternate value based on an analysis of available data. The conservative value is set as 10^{-3} failures per FH.

An analysis of failure rate data sources from [230, 231, 235] results in an alternate value being selected, although for the scope of the analyses in this thesis, only the order of magnitude is important to consider. Two additional failure modes for the RAT are identified; the first is the failure to deploy the RAT, and the second is a mechanical failure within the RAT's actuating mechanism or within the RAT's power drive components. A bottom-up approach was used to estimate the failure rate for the failure to deploy the RAT using a service bulletin providing maintenance advisory information on dealing with such incidents for the Bombardier Global 5000 aircraft [230].

The bulletin states that during an in-flight deployment, the RAT failed to fully extend even with two attempts at extension and provides corrective maintenance instructions to improve the reliability of the RAT. The key piece of information here is that the bulletin mentions that no other in-service events had been reported till the date of publication. As a result, it is possible to compute the overall failure rate by determining the number of flight hours completed by the Global 5000 aircraft, which amounts to 300 000 hours as of 2015 [231] (a conservative estimate), thereby resulting in a failure rate of $3.3*10^{-6}$ failures per FH. The value shown in table B.1 is an arbitrary estimate but maintains the same order of magnitude as determined using the bottom-up approach. The conservative value of 10^{-4} failures per FH is applied to case studies 1 and 2, while the remaining case studies use the alternate value of $3.3*10^{-6}$ failures per FH for the RAT component.

The failure rate value for the engine is based on engine rotor failure data for the CFM56 engine. It is treated as conservative since the data is from an analysis conducted in 1984 [229, p. 12]. Reliable data for aircraft electrical distribution networks is available from [228], although, for the case studies in 4.1, several different types of aircraft electrical networks are considered, which may feature different types of transmission voltages. In case studies 4.3.2 and 4.3.1, electrical power is required for both propulsion and secondary power consumers, while in case study 4.3.3, only secondary power applications are explored. As a result, a conservative failure rate of $5*10^{-5}$ failures per FH is selected and applied to all the case studies. Finally, the EBHA is assigned the same failure rate as that of an EHSA based on the assumption that, in normal operation, the EBHA operates in a similar manner to the EHSA as it uses hydraulic power supplied by the aircraft hydraulic system.

B.2 Path to FTA .opsa file syntax for the example shown in figure 3.25

```
<?xml version=’1.0’ encoding=’utf-8’?>
<open-psa>

    <define-parameter name="P1" unit="float">
        <float value="2.666E-6"/>
    </define-parameter>

    <define-parameter name="P2" unit="float">
        <float value="2.666E-6"/>
    </define-parameter>

    <define-parameter name="P3" unit="float">
        <float value="1E-4"/>
```

```
</define-parameter>

<define-parameter name="P4" unit="float">
    <float value="1E-4"/>
</define-parameter>

<define-parameter name="P5" unit="float">
    <float value="1E-4"/>
</define-parameter>

<define-basic-event name="E001">
    <label>Loss of S1</label>
    <parameter name = "P1"/>
</define-basic-event>

<define-basic-event name="E002">
    <label>Loss of S2</label>
    <exponential>
            <parameter name = "P2"/>
            <mission-time/>
    </exponential>
</define-basic-event>

<define-basic-event name="E003">
    <label>Loss of D2</label>
    <exponential>
            <parameter name = "P3"/>
            <mission-time/>
    </exponential>
</define-basic-event>

<define-basic-event name="E004">
    <label>Loss of D1</label>
    <exponential>
            <parameter name = "P4"/>
            <mission-time/>
    </exponential>
</define-basic-event>

<define-basic-event name="E005">
    <label>Loss of C1</label>
    <exponential>
            <parameter name = "P5"/>
            <mission-time/>
    </exponential>
</define-basic-event>

<define-gate name="P001">
```

```
        <label>Loss of System</label>
        <and>
            <gate name="P002"/>
            <gate name="P003"/>
        </and>
    </define-gate>

    <define-gate name="P002">
        <label>Loss of S1D1C1</label>
        <or>
            <basic-event name="E001"/>
            <basic-event name="E004"/>
            <basic-event name="E005"/>
        </or>
    </define-gate>

    <define-gate name="P003">
        <label>Loss of S2D2C1</label>
        <or>
            <basic-event name="E002"/>
            <basic-event name="E003"/>
            <basic-event name="E005"/>
        </or>
    </define-gate>

    <define-parameter name="None" unit="float">
        <float value="0"/>
    </define-parameter>

</open-psa>
```

Appendix C

ASSESS L2-M2

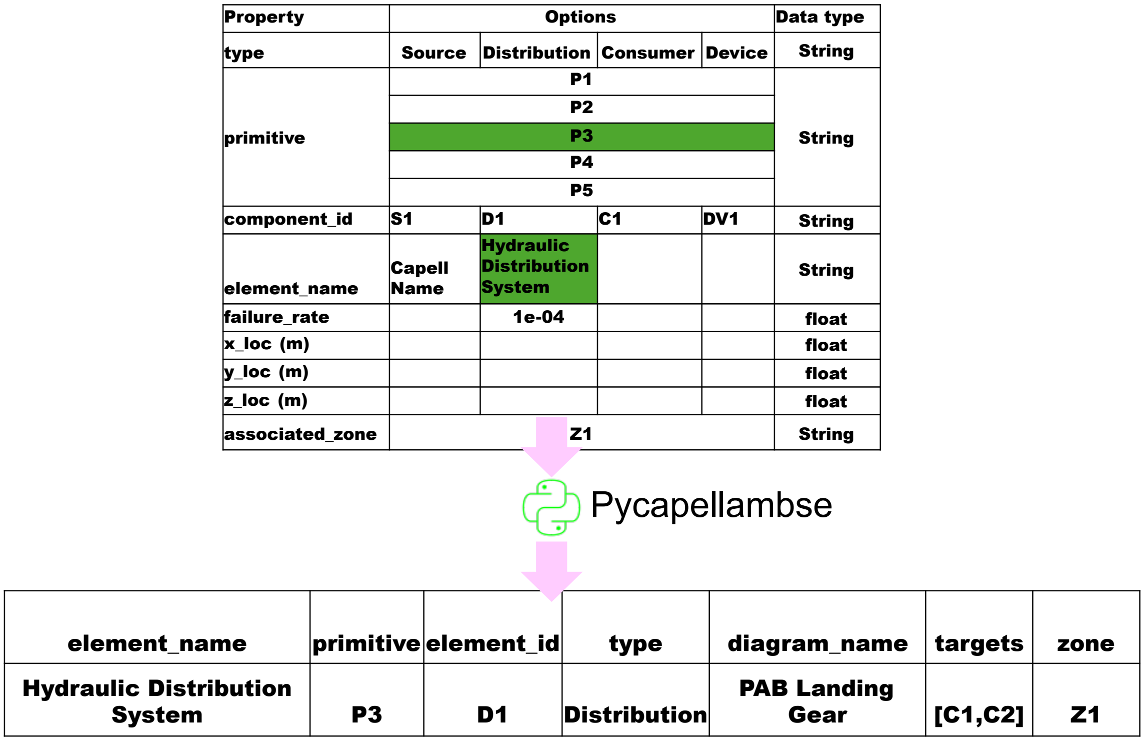
C.1 MBSE integration with conceptual PRA and conceptual ZSA tool

The ASSESS L2-M2 module enables zonal safety assessment and particular risk assessment to be carried out during the conceptual design stage with an initial system architecture and 3D model of the aircraft configuration. It allows for rapid analysis of system integration and aircraft configuration compatibility, and allows for early decision-making in defining system architecture placement within the aircraft or for recommending configuration changes due to safety considerations early in the design process. The ASSESS L2-M2 module is developed based on the methodology developed by Bamrah - a more detailed description of which can be found in [196]. This section outlines an interface between an MBSE specification and the input to the ASESSS L2-M2 module to enable zonal safety and particular risk assessment in aircraft conceptual design.

The input to the ASSESS-L2-M2 ZSA module consists of a list of components, their locations and the characteristics of the zones to which they are allocated. To enable a link between the architecture specification model in Capella and the ASSESS L2-M2 module, a set of PVMT fields are defined for elements in Capella. These fields help the architect specify what type of element, zone and conditions the component will be placed in. Furthermore, the architect can assign a primitive geometry to each component to help automate the generation of systems layout and installation of CAD geometries to perform conceptual PRA. The scope of this thesis, when it comes to ASSESS L2-M2, is limited to outlining an integration method to the generic element descriptor and the formal model-based system architecture specification.

Generic element descriptor & MBSE integration method

The property value management toolkit add-on is chosen to facilitate the link between the model-based architecture specification and the conceptual particular risk assessment (cPRA) and conceptual zonal safety assessment (cZSA) environment, which also interface with a spreadsheet and a parametric aircraft geometry modeller (OpenVSP). A PVMT property list is created to capture all the relevant information that is required by the cPRA and cZSA analyses. To support the need to have a preliminary geometry of the system component, a set of primitive shapes such as a box, cylinder and sphere are defined to represent the volume envelope of a system component and are designated specific identifiers as shown in figure C.1. The system architect can select the appropriate primitive, and also specify in which aircraft zone the component is to be placed. Alternatively, the architect may also specify directly the coordinates of the desired system placement.



Property	Options				Data type
type	Source	Distribution	Consumer	Device	String
primitive	P1				String
	P2				
	P3				
	P4				
	P5				
component_id	S1	D1	C1	DV1	String
element_name	Capell Name	Hydraulic Distribution System			String
failure_rate		1e-04			float
x_loc (m)					float
y_loc (m)					float
z_loc (m)					float
associated_zone	Z1				String

element_name	primitive	element_id	type	diagram_name	targets	zone
Hydraulic Distribution System	P3	D1	Distribution	PAB Landing Gear	[C1,C2]	Z1

Figure C.1: Outline of the process for integrating MBSE with the cZSA and cPRA tools

Similarly, to support some features of the cZSA, information about each zone can be explicitly specified within the model-based architecture specification. Figure C.1 shows the typical information required to be specified for each component. The link between the architecture specification model and the geometrical modeller is established by extracting the information from the model using the pycapellambse module and transferring it to an intermediate format that will enable the architect to manually install the system components in the geometric modeller. The information required for the cZSA can be automatically transferred to the input spreadsheet of the cZSA tool.

Since the nodes of the generic element descriptor implemented in Networkx can also store information as node properties, the same approach is applicable to be able to directly link the graph to the cZSA and cPRA tools. Future development will enable complete automation between the MBSE specification and the geometric modeller.

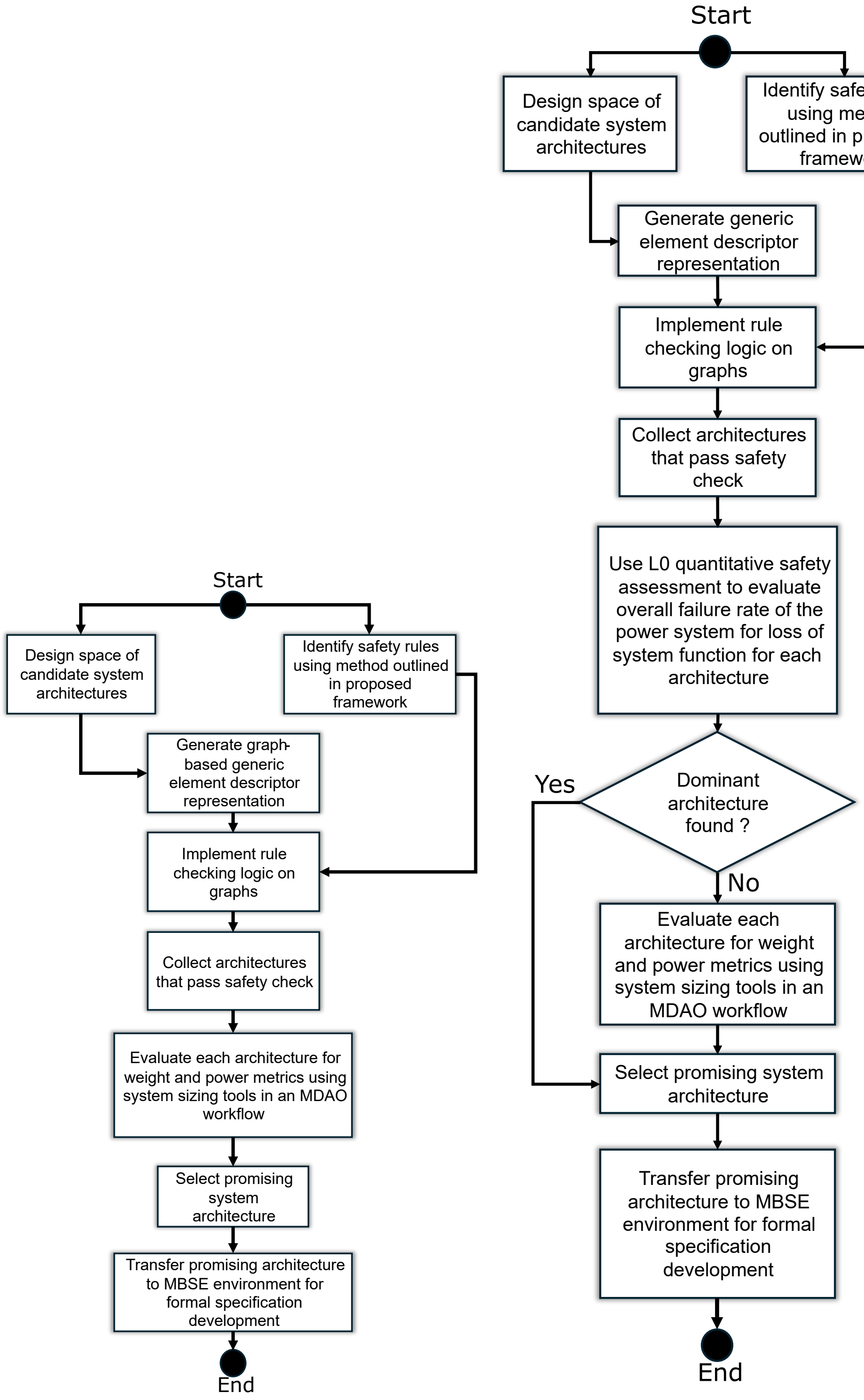
Appendix D

Framework Implementation

D.1 Approaches to Implement Elements of the Safety-Focused Systems Architecting Framework in Aircraft Conceptual Design

This section prescribes how the safety-focused systems architecting framework for aircraft conceptual design can be applied in practice. Note that this section does not present an exhaustive list of ways in which the framework can be applied; rather, it outlines how the framework is anticipated to be used for systems architecting in an academic and industrial context.

Figure D.1(a) shows a simple approach where a design space of candidate architectures is evaluated for safety using a set of safety rules that are identified for each system under consideration. The candidate architectures are represented using the graph-based generic element descriptor, and the safety rules are applied as logical checks of the connections between the different elements of the generic element descriptor. The architectures that pass the safety checks are evaluated in an MDAO workflow to determine the impact of the architecture at the aircraft level. The input required by the system sizing tools is obtained from information stored in the system architecture descriptor. Once a promising architecture is identified, it is then transferred to an MBSE environment for detailed development.

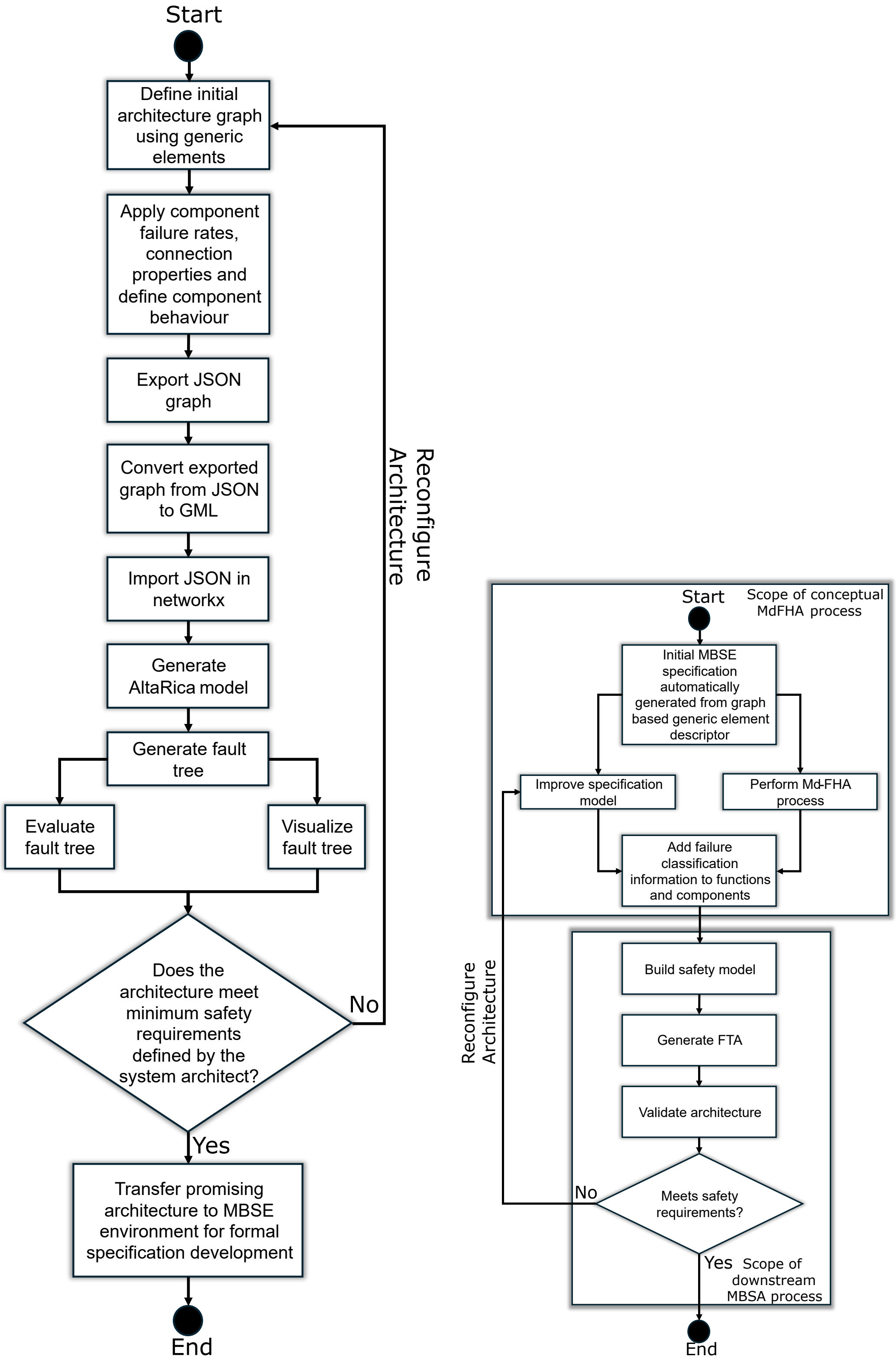

Start
Design space of candidate system architectures
Identify safety rules using method outlined in proposed framework
Generate graph-based generic element descriptor representation
Implement rule checking logic on graphs
Collect architectures that pass safety check
Evaluate each architecture for weight and power metrics using system sizing tools in an MDAO workflow
Select promising system architecture
Transfer promising architecture to MBSE environment for formal specification development
End
Start
Design space of candidate system architectures
Identify safety rules using method outlined in proposed framework
Generate generic element descriptor representation
Implement rule checking logic on graphs
Collect architectures that pass safety check
Use L0 quantitative safety assessment to evaluate overall failure rate of the power system for loss of system function for each architecture
Dominant architecture found ?
Yes
No
Evaluate each architecture for weight and power metrics using system sizing tools in an MDAO workflow
Select promising system architecture
Transfer promising architecture to MBSE environment for formal specification development
End

(a) Process for rule-based safety assessment and integration with MDAO

(b) Process for rule-based safety assessment, quantitative safety evaluation, and integration with MDAO

Figure D.1: Overview of two different approaches to applying the safety-focused systems architecting framework for aircraft conceptual design

Figure D.1(b) incorporates quantitative safety evaluation into the process outlined in Figure D.1(a). The architectures that pass the safety checks are further evaluated using L0 quantitative safety evaluation outlined in section 3.6.2, and the overall system failure rate for loss of system function is used to compare architectures and determine if a dominant architecture exists. If a promising architecture is found, then it can be automatically transferred to an MBSE environment, and if not, then each architecture is evaluated for its impact at the aircraft level using an MDAO workflow. A combination of metrics, including overall system failure rate for loss of system function, along with aircraft-level metrics such as MTOW, fuel burn and system power consumption, can be used to select an architecture that meets the requirements. The selected architecture is then transferred to the MBSE environment for detailed specification development.


Start
Define initial architecture graph using generic elements
Apply component failure rates, connection properties and define component behaviour
Export JSON graph
Convert exported graph from JSON to GML
Import JSON in networkx
Generate AltaRica model
Generate fault tree
Evaluate fault tree
Visualize fault tree
Does the architecture meet minimum safety requirements defined by the system architect?
No
Reconfigure Architecture
Yes
Transfer promising architecture to MBSE environment for formal specification development
End
Start
Scope of conceptual MdFHA process
Initial MBSE specification automatically generated from graph based generic element descriptor
Improve specification model
Perform Md-FHA process
Add failure classification information to functions and components
Build safety model
Generate FTA
Validate architecture
Reconfigure Architecture
No
Meets safety requirements?
Yes
Scope of downstream MBSA process
End

(a) Process for interactive safety-focused systems architecting and architecture evaluation using FTA

(b) Process for Model-driven Functional Hazard Assessment in aircraft conceptual design

Figure D.2: Overview of processes for interactive safety-focused architecting and a Model-driven Functional Hazard Assessment

Figure D.2(a) outlines the interactive safety-focused systems architecting approach that is driven by the graph-based generic element descriptor and the automatic generation of an AltaRica safety model as described in section 3.6.2. The system architect first develops the architecture using a graphical interface, such as Arrows, and then exports the graph, which is then used to generate the safety model in AltaRica. The AltaRica model is used to generate a fault tree, which is then evaluated to generate safety metrics such as the failure rate for specific failure conditions. The cardinal failure conditions described in this thesis are included in the safety model by default. However, the architect may identify their own failure conditions of interest from a system FHA and modify the AltaRica model as required.

The architect then inspects if the architecture meets safety requirements for all the evaluated failure conditions and determines if any modifications are required. If the architect deems that changes need to be made, then the process starts again with the export of the graph-based descriptor. If the architect is satisfied, then the system architecture is automatically transferred into an MBSE environment for further development.

Figure D.2(b) outlines the MdFHA process that is carried out under two specific scenarios. The first is when an initial system architecture specification is instantiated with information from the graph-based generic element descriptor. The second is when the architect starts to model a system architecture directly within an MBSE environment. In both scenarios, the architect follows the MdFHA process outlined in section 3.6.1 and enriches the system model with information about failure conditions, classification of failure conditions and additional safety requirements such as the budgeted failure rate for each system function. These activities lie within the scope of the MdFHA described in this thesis and prepare a system architecture specification that can be further developed and used to automatically generate a safety model with which the architecture can be validated. The architecture can then be modified as required according to the outcomes of the fault tree assessment. The details of the MBSA process are described in [236].