33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

# SYSTEMS ARCHITECTING: A PRACTICAL EXAMPLE OF DESIGN SPACE MODELING AND SAFETY-BASED FILTERING WITHIN THE AGILE 4.0 PROJECT

Andrew. K. Jeyaraj[1], Jasper H. Bussemaker[2], Susan Liscouët-Hanke [3] & Luca Boggero[4]

[1 & 3] Concordia University, Department of Mechanical, Industrial and Aerospace Engineering, Montreal, Canada
[2 & 4]DLR, Institute of System Architectures in Aeronautics, Hamburg, Germany

## Abstract

The aerospace industry strives toward innovative aircraft concepts that feature increasing electrification to meet environmental and business targets. Advanced Multi-Disciplinary Design Analysis and Optimization (MDAO) frameworks have been developed to help evaluate these aircraft and their systems. However, the system architecting process still relies on a system architecture baseline from past aircraft programs or historical data, thereby precluding the exploration of a larger design space and identifying optimal solutions for further development. Furthermore, the evolution of system safety is a critical factor in establishing the feasibility of a system architecture solution. Therefore, there is a need to explore a large design space of system architectures for safety, certification, and performance requirements in an efficient manner. This paper presents a rule-based safety assessment approach within a systems architecting framework that demonstrates the ability to generate and filter a large design space based on safety heuristics. This approach is demonstrated using a case study for an aircraft landing gear braking system.

**Keywords:** system architecture, design space exploration, design space modeling, safety assessment, conceptual design.

## 1. Introduction

Exploring multiple aircraft concepts at the conceptual design stage in the aircraft development process is critical to identifying the best fit for a given set of requirements. This process is called design space exploration and is typically carried out at the aircraft level, reflecting choices in engine placement, wing configuration, and placement. However, typically only a few aircraft system architectures[1] are explored as they add significant complexity to the design problem and require detailed knowledge about many subsystems. This is not ideal as a large design space of system architecture options is excluded, and a potentially inefficient fit must be made between a baseline architecture and the aircraft level configurational choices. A shift towards systematic design space exploration methods coupled with a physics-based evaluation of architecture candidates is required to improve the aircraft development process.

Recent advances in system architecting have focused on using physics-based methods for evaluating system architecture options. Liscouët-Hanke in [1] & [2] demonstrates a power-based framework for system architecture evaluation. Similarly, Chakraborty [3] and Lammering [4] also introduce system architecture and sizing and performance evaluation frameworks. Although these techniques are generic in formulation, they are demonstrated on more electric systems architectures that are based on an underlying conventional baseline. Furthermore, a large number of architecture variants are not explored, and unconventional architectures are typically not considered.
Efforts have also been made to develop Multidisciplinary Design Analysis and Optimization (MDAO)

[1] Sometimes also called “onboard system”; here, the authors address system architecting as the definition, documentation and evaluation of aircraft system architectures such as flight control system, electrical or hydraulic power systems.

capabilities that also integrate system-level analyses. In the AGILE project[2], system-level analyses that consider increasing levels of system electrification are included in an MDAO workflow [5]. These are based on the system architecture evaluation methods developed by Boggero et al. [6]. The follow-up project called AGILE 4.0[3] focuses on integrating system architecting by adopting a Model-Based Systems Engineering (MBSE) approach - from stakeholder definition and requirements modeling to architecture design space modeling capabilities to support the design process [7]. The architecture modeling capabilities are implemented in the ADORE (Architecture Design and Optimization Reasoning Environment) tool, developed by the German Aerospace Research Center DLR [8].

The system architecture design space exploration process has benefitted from improved architecture generation and evaluation methods. However, searching through a design space based on performance evaluation alone can be intractable due to a large number of available combinations and potentially significant time needed to evaluate one candidate design. Bauer et al.[9] use a set of constraints that are based on technological choices and design practices to filter a design space of conventional flight control systems architecture and use the Airbus A340 roll control system architecture as a case study. Safety constraints are applied as a black box function that evaluates the degradation of roll performance for specific failure cases. Zeidner et al.[10] and Becz et al.[11] apply abstraction within a platform-based design framework to enable design space exploration by exploring the interconnections between system architecture elements. Zeidner et al. [10] further introduce the concept of configurational filtering, which is then used in conjunction with performance-based evaluation to reduce the complete design space to a set of feasible architectures. Zeidner et al. also emphasize the efficacy of using a generative filtering approach over evaluative filtering in arriving at a set of feasible architectures[10].

Garriga et al. in [12] & [13] 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, such as the nature of power sources, such as primary or backup, and the variation of minimum power source allocation and redundancy requirements, are not considered. Furthermore, 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. [14], [15] 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 AGILE4.0 project. However, methods to filter the design space to ensure that a remaining set of system architectures are implicitly safe are yet to be developed. Initial work on architecture definition based on heuristics by Chakraborty and Mavris [16] is promising. The development of a generic set of safety heuristics would be beneficial to filter through a large design space. This has been developed in the "Aircraft Systems Safety assESSment" (ASSESS) tool, which implements a set of generic safety heuristics on a system-by-system basis. Furthermore, it provides a simplified means of representing a system architecture in such a way that the safety rules can be quickly evaluated and enable the filtering of a large design space.

[2] AGILE: Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts: https://www.agile-project.eu/
[3] AGILE 4.0 is a follow up to the AGILE project to develop cyber-physical methods for aircraft development: https://www.agile4.eu/

Jeyaraj and Liscouët-Hanke [17] present a unified safety-focused system architecting framework that addresses each stage of the systems architecture design space exploration process, namely: architecture definition, architecture description, and architecture evaluation shown in Figure 1. In their approach, the system architecture definition phase is characterized by the population of a large design space of system architecture options which are then selectively filtered based on a set of safety heuristics.

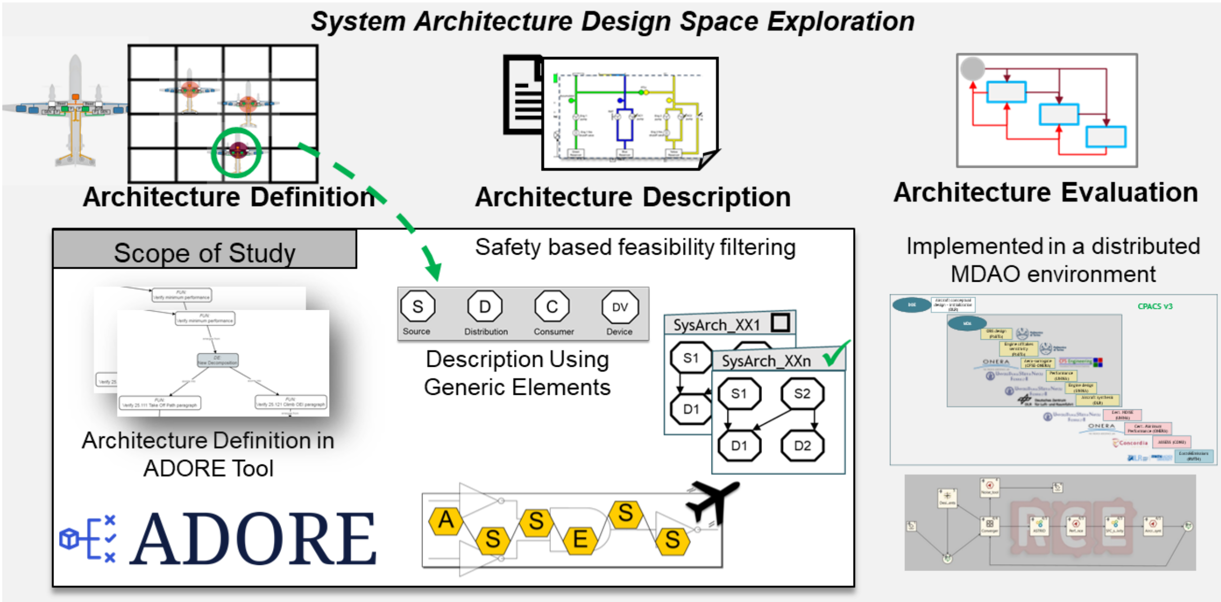



*Figure 1: Design space exploration activities and corresponding means of implementation in this study*

The architecture description phase involves the development of system architecture specification models in an MBSE environment, which are then evaluated in an MDAO workflow during the evaluation phase.

The safety heuristic-based system architecture filtering approach implemented in the ASSESS tool and the aspects presented in this paper adds to state of the art along the following points of novelty:

1. The introduction of a generic or abstracted means of representing system architectures that allows safety heuristics to be evaluated effectively and enables exploration of novel or non-intuitive system architectures
2. The prescription of safety heuristics that consider the nature of power sources, power distribution, power consumption, and the incorporation of control elements within the chain of power.
3. The establishment of a basis for safety heuristics in certification regulation along with industry derived best practices and existing system architecture implementations

This paper presents a practical example of system architecture design space exploration that covers the system architecture definition and description phases using the design space modeling capabilities of the ADORE tool and the rule-based safety filtering approach implemented in the ASSESS tool. The landing gear braking system is used to demonstrate the process of generating a design space of landing gear braking system architecture options. These options vary in the nature of actuation technology that is used on the braking device. The architectural options are then filtered based on conformity to a predefined set of safety heuristics. Filtering the architectural options ensures that only feasible architectures are evaluated using physics-based approaches, making the architecting process more efficient.

## 2. Workflow Description

This section outlines the workflow used within the AGILE 4.0 project to develop and filter the design space of system architectures which are then evaluated using system analysis tools. The key tools for design space modeling, ADORE and ASSESS, are described below.

### 2.1 ADORE (Architecture Design and Optimization Reasoning Environment)

The implemented architecture design space modeling methodology is based on the Architecture

Design Space Graph (ADSG) [18]: a directed graph with nodes representing architecture elements like functions and components and edges representing derivation relationships (e.g., a function derives the existence of a component). This graph can be used to identify architectural decisions and generate architecture instances by making these decisions. Numerical optimization problems can be formulated by mapping decisions to design variables and performance metrics to objectives and constraints.

ADORE implements the ADSG in a Python environment coupled with a web-based graphical user interface for editing and inspecting the design space model [8]. Figure 2 shows a jet engine-architecture design space model. In this model, the top-level function "Provide propulsive power" drives the derivation of architectures shown by directed edges. Blue-dashed edges represent architectural decisions with mutually exclusive options, for example, whether to use the fan or nozzle component to fulfill the "Accelerate air" function. Bi-directional red lines link incompatible elements.

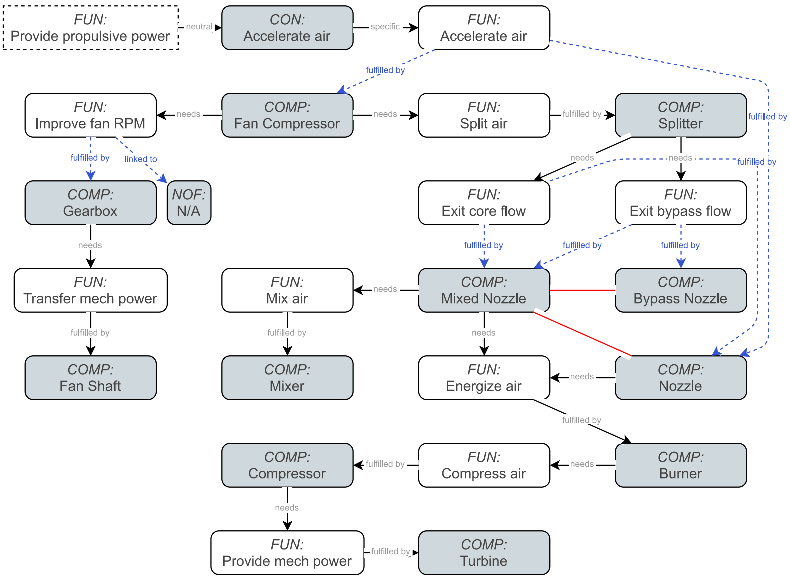



*Figure 2: Example jet engine architecture model created using ADORE, from* [8].

Additionally, ADORE offers interfaces to various Python-based optimization frameworks, which enables using existing optimization algorithms to suggest new design vectors to evaluate. These design vectors can then be converted to architecture instances for evaluation, taking into account the design variable hierarchy (see [18] for more details). No architecture evaluation capabilities are implemented in ADORE. Instead, it is up to the user to implement architecture evaluation by connecting an evaluation tool or workflow to the architecture generation process and providing feedback to the optimizer in terms of objectives and constraints.

## 2.2 ASSESS (Aircraft Systems Safety assESSment)

The conventional aircraft safety assessment follows the SAE ARP4761 [19], which consists of several steps at the aircraft and system levels and is a key activity in the certification of an aircraft. This process is typically done manually over various stages in the aircraft development process. The ASSESS tool implements these safety assessment processes in a manner that is conducive to system architecting at the conceptual design stage. The ASSESS tool consists of two modules, L0 and L1. The L0 module introduces a rule-based safety assessment using safety heuristics derived from the following sources: 1) analysis of existing system architectures, 2) pertinent certification regulations corresponding to required redundancies in power generating, power distribution, and power-consuming system architecture elements, 3) established literature and best practices (e.g., by Chakraborty and Mavris [16]) and 4) industry derived best practices in specifying system architectures. This study will focus on filtering a design space of landing gear braking systems architectures using safety heuristics derived for Part-25[4] and Part 23 aircraft.

[4] In this paper, the abbreviation “Part 25” is used, referring to the Part 25 of the airworthiness standards for transport category airplanes (e.g. in Canada: Airworthiness Manual Chapter 525)

The authors (1&3) extensively analyzed more than 30 different aircraft systems architectures. The cohort of aircraft was divided based on certification basis into the Part 23[5] and Part 25 based categories. System schematics from training, flight, and maintenance manuals were used as sources in addition to academic literature and supplier documentation. Initially, the authors considered the schematics and sources on a system-by-system basis. Electrical, Hydraulic, Flight Control, and Landing Gear Braking systems- to name a few- were analyzed in isolation to determine typical redundancies in power distribution and generation systems and allocation of these distribution sources to power-consuming systems.

In addition to the allocation of power distribution sources to power consumption elements, the authors identified that in the case of electrified aircraft systems, there is a component of electrical control involved that requires power supply- the absence of which can have an equivalent impact on a system function as the loss of primary sources of power or a distribution system. Therefore, it is essential to consider the incorporation of electrical controllers as analog to conduits of power, especially in the signaling of electrical actuation functions.

The safety heuristics that are evaluated in each architecture are based on literature and a survey of existing system architecture implementations. The allocation of power systems to actuation elements is also studied to determine a pattern in the type and the minimum number of required allocations. Additionally, certification rules are used to determine the minimum number of power systems that supply a given brake actuation element. In addition to power allocation, control allocation was also investigated. This is important in the case of electrically signaled braking architectures where both power and control influence the braking function and require redundancies.

In ASSESS L0, the authors (1&3) introduce a set of generic elements that are used to represent a system architecture. These elements are used to describe the architecture and simplify the process of rule evaluation. The generic elements are shown in Figure 3, and each system architecture is comprised of connections between these elements.

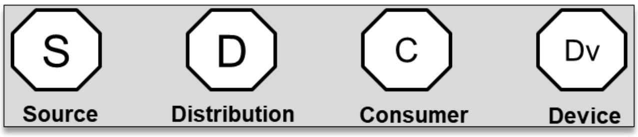



*Figure 3: Generic elements used to represent system architectures*

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 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 important 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.

[5] In this paper, the abbreviation "Part 23" is used, referring to the Part 23 of the airworthiness standards for normal category airplanes (e.g. in Canada: Airworthiness Manual Chapter 523)

*Table 1: Nomenclature of generic elements used to represent architectures in the ASSESS tool*

| Generic Element | Referenced Nomenclature | Physical Components | Associated Elements |
|---|---|---|---|
| Source | Power Generation Systems[1] Prime Movers [16] | Engine, Battery | Fuel |
| Distribution | (PDS)[1],(MPGDS,HPGDS,EPGDS) [16] | Hydraulic, Electrical Distribution System | EDP[6], EMP[7], IDG[8] |
| Consumer | Power Consuming Systems[1],Subsystem (FCS, ECS,LDG)[16] | Flight Control System, Flight Control Actuators, Landing Gear Braking Devices | EHA[9], EMA[10], EHSA[11] |
| Device | - | Wheels, Control Surfaces | - |

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. Elements of control that fall within the power path, such as controllers, are also accommodated as "Consumer" elements within this nomenclature. Additionally, intermediate elements may also be specified, but a description of these is beyond this paper's scope.

## 2.3 ADORE-ASSESS Link

The ADORE tool generates a large number of architecture instances from the design space model. This design space will be filtered for safety by the ASSESS tool. ADORE and ASSESS represent architectures in different formats according to their respective modeling purposes, and so a translation step is needed [20]. The passing of information between ADORE and ASSESS can be accomplished using a common data language or a direct transfer, as shown in Figure 4. The usage of a common data language, as would be appropriate in large-scale collaborative MDAO processes, is described in [8]. This study applies the direct transfer using class factories [21]. Class factories offer a convenient way to extract relevant information from generated ADORE system architectures: class factories represent rules for instantiating custom Python objects for every linked ADORE element encountered in a specific architecture instance. For example, a class factory might define the rule: instantiate the `Propeller` class for every "Propeller" component instance and set the `diameter` property to the value of the "Diameter" attribute of the component instance.

[6] Engine Driven Pump
[7] Electric Motor Pump
[8] Integrated Drive Generator
[9] Electro-Hydrostatic Actuator
[10] Electro-Mechanical Actuator
[11] Electro-Hydraulic Servo Actuator

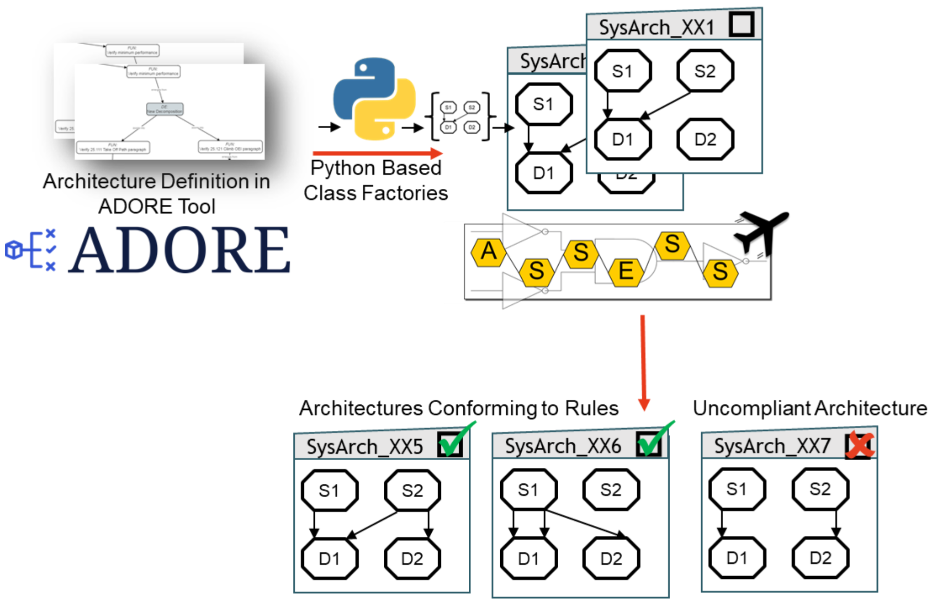



*Figure 4: ADORE- ASSESS interaction*

## 3. Case Study: Landing Gear Braking System Architecture

### 3.1 Case Study Description

The case study is focused on generating a large design space for aircraft landing gear braking system architectures using the ADORE tool and then filtering the resulting architectures for feasibility based on safety heuristics. This will determine whether an architecture in the design space conforms to a set of heuristics that represent implicit system safety. In this case study, no feedback will be provided to ADORE for guiding the design space exploration, which is something that could be realized in other workflows by evaluating design objectives.

In the AGILE 4.0 project, several application cases focus on evaluating safety, certification, and maintenance constraints in an MDAO workflow. A regional aircraft is used for these studies, which include evaluating aircraft electrification and retrofitting a hybrid electric powertrain. The top-level aircraft level aircraft requirements for these aircraft are at the border between Part 23 and Part 25 certification regulations and provide a favorable testbed for observing the impact of certification regulations on the overall aircraft design. Within this subset of top-level aircraft requirements, this study focuses on the landing gear braking system, as it represents a system characterized by choices in actuation technology that results in the need for a power system, which, in turn, requires redundancies in selection and allocation. Furthermore, landing gear braking system requirements also vary depending on the set of applicable certification rules (Part 25 or 23). The workflow used in this study is illustrated in Figure 5.

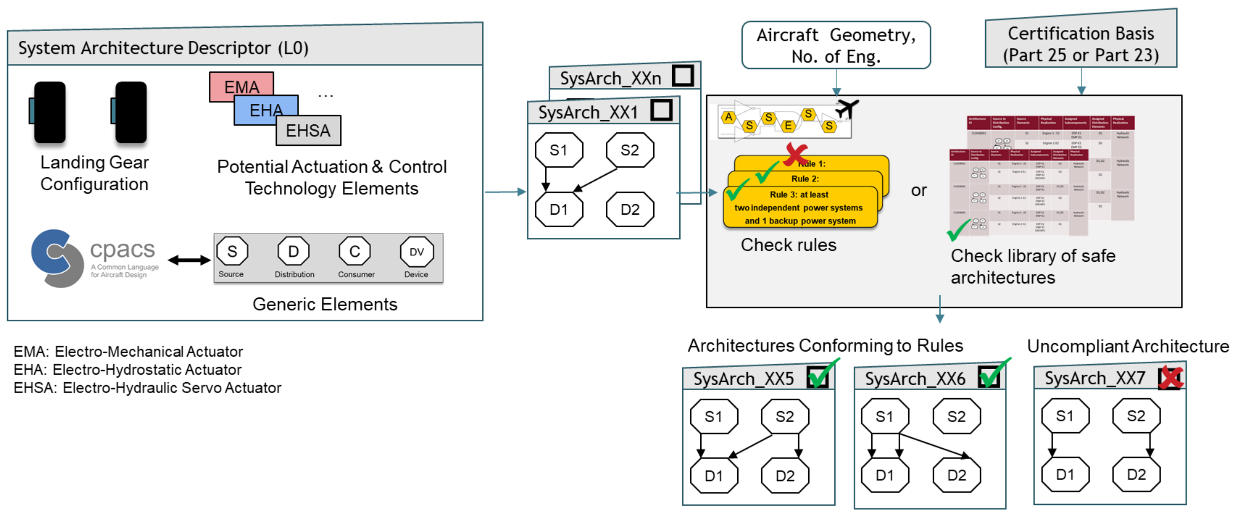



*Figure 5: Case Study Description: Landing gear braking system with brake actuation technology choices evaluated for safety rules in the ASSESS tool*

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 device. Each braking unit can comprise an electro-hydraulic servo actuator (EHSA), electromechanical actuator (EMA), and local hydraulic generation-based actuator. Therefore, each gear can also 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 EMA.

## 3.2 Design Space Definition in ADORE

This study covers two important aspects of systems architecting: (1) design space definition and architecture generation (using ADORE) and (2) design space evaluation/filtering (using ASSESS). The first step involves creating a design space of landing gear braking system architectures in ADORE, which will be described in this section. These candidate system architectures are then filtered using the generic safety heuristics in ASSESS, which will be described in section 3.4. The design space generation process in adore requires an initial system architecture model to be defined. This requires the identification of functions and components to fulfill those functions and also to provide 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 6. 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 shown in section 2 of Figure 6. Power is generated by the engine or emergency power system 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).

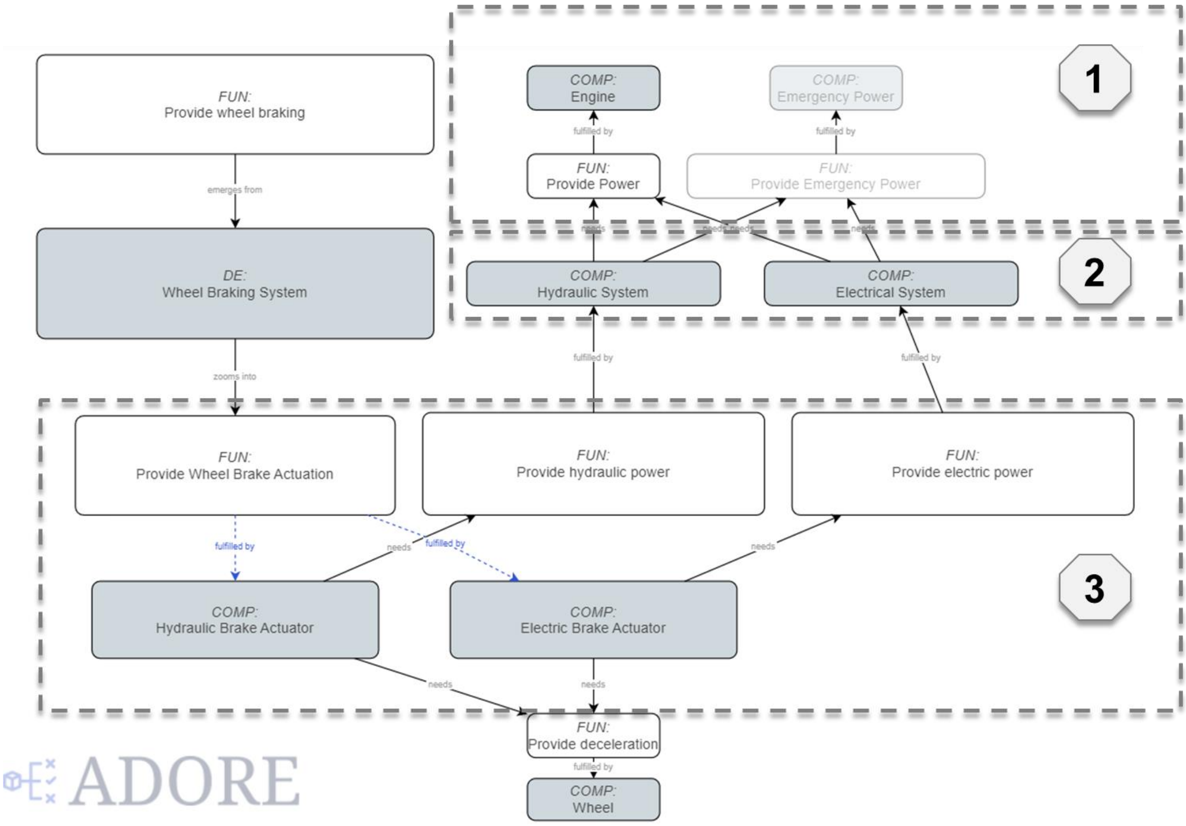



*Figure 6: System architecture definition in ADORE*

Next to 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.

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, respectively. 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 7. Each hydraulic system needs to be connected to at least one source: their input ports accept between 0 and 2 connections from the engines, as 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

## 3.3 Translation from ADORE to ASSESS Architectures

The ADORE design space model is used to generate different architectures automatically. A generated architecture instance represents a combination of choices between electric or hydraulic actuation and a specific connection pattern from sources, through distributions, to consumers (as required by ASSESS).

In ASSESS, a system architecture is defined using a directed graph containing the node types discussed in previous sections. Translation from the ADORE to the ASSESS format is implemented using class factories [21]. Five class factories are defined: one for connection ports and four for each type of ASSESS node. The port class factory is needed to extract connection information from the architecture model. The four-node class factories represent each ASSESS node type (Source, Distribution, Consumer, Device) and are linked to the respective components in the ADORE model. These factories instantiate a `GraphNode` class with the following properties:

- Prefix: a string representing the ASSESS node type;
- Connection targets: a list of `GraphNode` objects connected to this node.

.

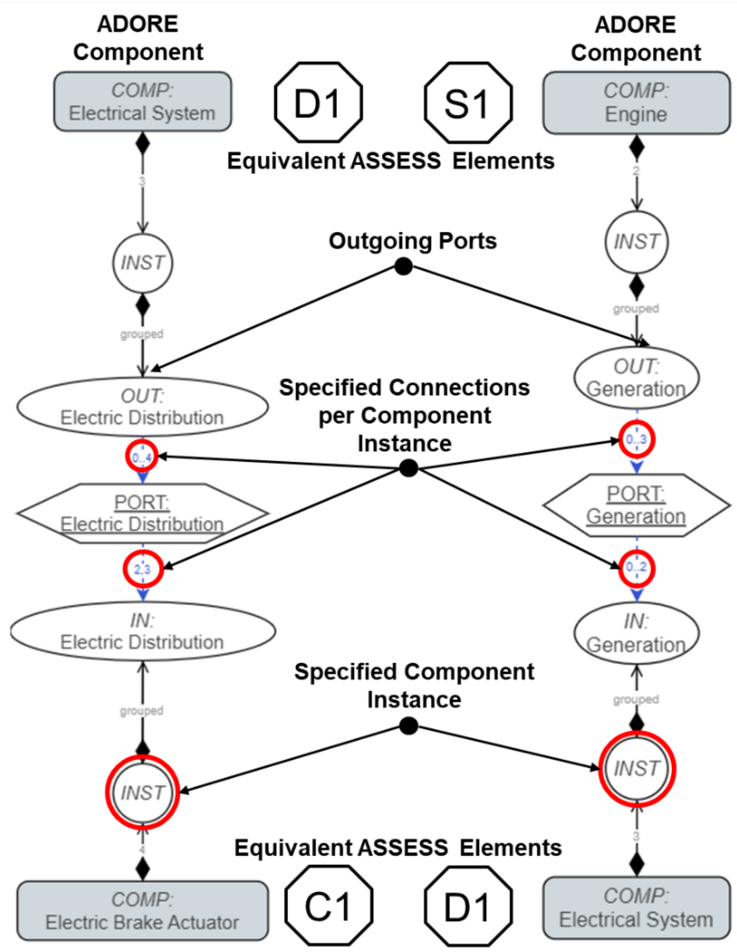



*Figure 7: Port connections definition in ADORE and mapping to ASSESS elements*

This way, a component representing an engine that is connected to a hydraulic system would result in the instantiation of a `GraphNode` that contains each of the `GraphNode` objects representing the hydraulic systems as its connection targets. Finally, the directed graph needed for ASSESS is then created by adding a node and relevant edges to a directed graph for each instantiated `GraphNode` object.

## 3.4 Safety-heuristics based filtering

The certification basis in Part 23 and Part 25 Aviation Regulations was analyzed to extract safety heuristics that are applied to power generation and distribution systems. Comparing these regulations with existing system architecture implementations allows ambiguities to be resolved and a generic heuristic developed. For instance, in the case of landing gear systems, it was identified that Part 25 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 Part 23 certification basis, the authors noted that only a single primary source of hydraulic power was required, along with a backup source. Here, the authors distinguish between *main/primary*, *alternate*, and *backup* 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 handpump to an independent backup power source such as an APU-driven pump or generator.

Below, a sample of four rules is presented for the wheel braking actuation system:

- **Rule 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 also derive from an independent power source or be supplied locally (using an accumulator or human-powered supply).
    - 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 provisioned 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 sources elements. In this case, two independent electrical systems will suffice.
    - 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 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 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 4: Controller redundancy:**
    - For electrical braking system architectures, one brake control unit is to be allocated for every inboard-outboard wheel pairing
    - Two sub-controllers must be allocated to each wheel pair for every brake control unit
    - At least two electrical distribution elements must supply each controller

*Table 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 2 shows how the proposed heuristics are related to specific certification regulations. CFR 25.1309 and 23.2510 prescribe safety requirements for the proper functioning of installed systems under any foreseeable operating condition. These further impose safety requirements that ensure that any failures leading to an unsafe condition of the aircraft are extremely improbable. Under the assumption that the ARP 4761 process is used to demonstrate compliance to these regulations and that the probabilistic approach of safety assessment techniques tends to prescribe increasing system redundancy [22]- 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 also 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.

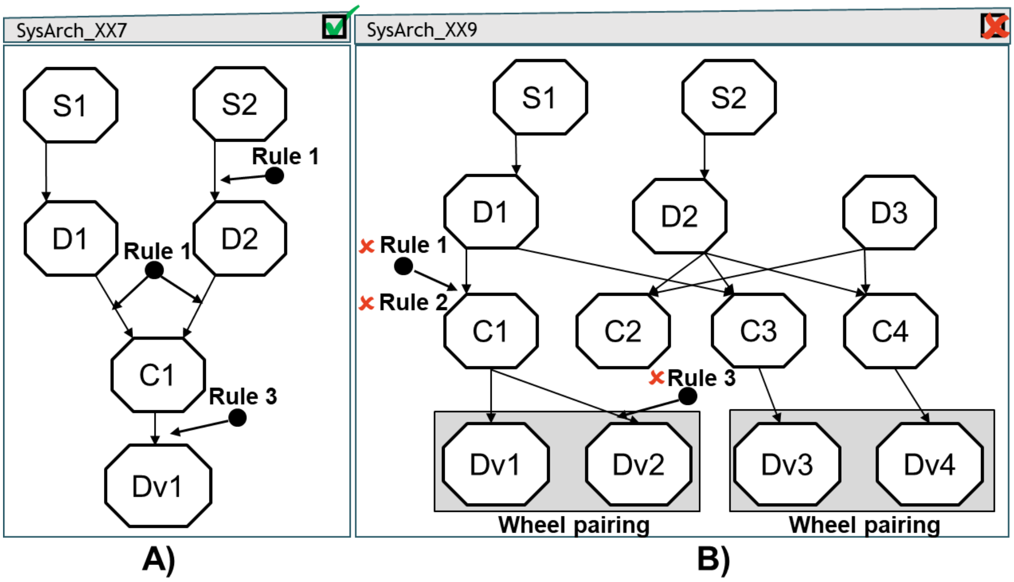



*Figure 8: Example of rule evaluation on sample landing gear braking (conventional) system architecture*

Figure 8 illustrates examples of the rule evaluation within ASSESS. In Figure 8A), the connections between sources (S1 & S2) and distribution ( D1 & D2) are such that the consumer (C1) is always supplied with two independent systems – since S1 and S2 are independent primary sources. The consumer C1 is allocated to a unique device, Dv1, and thus satisfies rule 3. On the other hand, Figure 8 B) shows a case that fails checks for compliance with Rule 1, 2, and 3. Here the consumer C1 is only supplied by one independent primary distribution, D1, and this further 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.

## 4. Results

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 those for the electric landing gear braking. Finally, a specific case of building a design space using safety heuristics is shown.

### 4.1 Design of Experiments (DOE)

The python library 'pymoo' [23] 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 varied during the DOE. These are as follows:

1. The connection between engine and distribution system which accounts for 49 variations
2. The connection between the emergency(backup) generation system and distribution system 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. The results are included in the "DOE Size" column of Table 3 and Table 4. Hydraulic (Conventional) Landing Gear Braking

*Table 3: Rule-based design space filtering results for hydraulic (conventional) landing gear braking system architecture*

| 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 3 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.

### 4.2 Electrical Landing Gear Braking Case

Table 4 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 the hydraulic case, as the requirements for three main systems allocated per consumer element are more stringent than for the hydraulic case. Furthermore, ensuring the independence of electrical systems also requires an independent backup to be supplied to at least one distribution element.

*Table 4: 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 | Part 25 | 86 | 14 | 14.00 |
| | 100 | Part 23 | 60 | 40 | 40.00 |
| **5000** | 4873 | Part 25 | 4250 | 623 | 12.78 |
| | 4889 | Part 23 | 2950 | 1939 | 39.66 |
| **10000** | 9513 | Part 25 | 8288 | 1225 | 12.87 |
| | 9489 | Part 23 | 5678 | 3811 | 40.16 |

## 4.3 Constrained Design Space Analysis

The 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 problem 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.

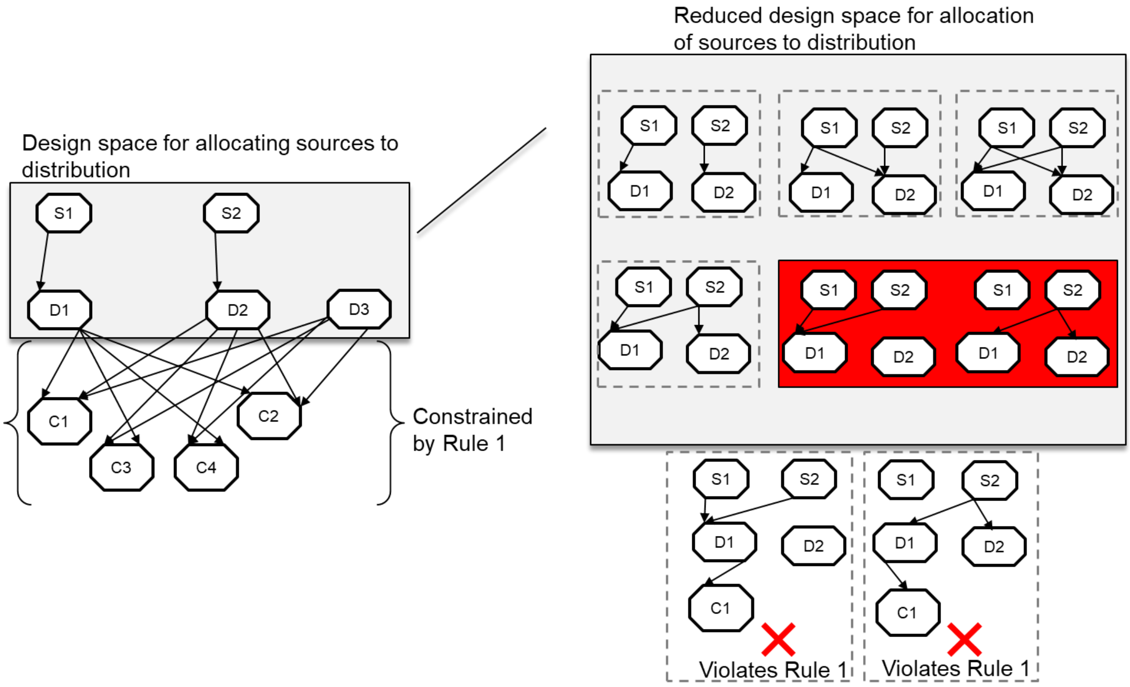



*Figure 9: Generation of a constrained design space*

Finally, the connections between source and distribution are enumerated and shown in Figure 9 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 5 for the hydraulic braking system case. A similar case is observed for the electric landing gear braking case as well. 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 6 shows the reduced list of architectures conforming to the rules.

*Table 5: Architectures generated based on safety rules for conventional hydraulic brake actuation*

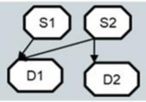
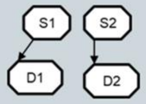
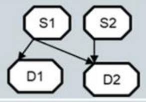
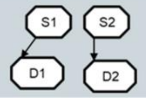
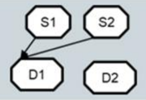
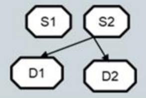
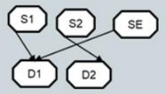
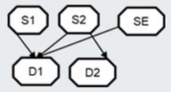
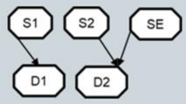
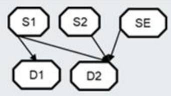
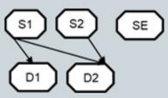
| Architecture ID | Source to Distribution Config. | Source Elements | Physical Realization | Assigned Subcomponents | Assigned Distribution Elements | Physical Realization |
|---|---|---|---|---|---|---|
| CLDGB001 ✓ |   | S1 | Engine 1: E1 | EDP-E1, EMP-E1 | D1 | Hydraulic Network |
| | | S2 | Engine 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 6: 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 | - | |

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 5 and Table 6. Source elements are allocated individual engines, which are further allocated engine-driven pumps, and distribution elements are identified as electrical or hydraulic networks.

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, then the overall complexity of the design space can be reduced by choosing to focus on specific systems for complete enumeration of their respective design spaces and using the constrained design space approach for the remainder.

## 5. Conclusion and future work

This paper presents a practical approach to modeling and filtering large design spaces of candidate system architectures using safety heuristics. A link between the system architecture design space modeling tool ADORE and the rule-based safety assessment module of the ASSESS tool was created. A set of safety heuristics derived from system architecture analysis, certification regulation, and industry best practices were formalized and used to filter a design space of conventional and electric landing gear braking system architectures. These 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. Currently, the filtering process is applied on a system-by-system basis. Future work will involve testing the safety heuristics on novel system architectures and applying the generic heuristics to multiple systems concurrently.

## 6. Contact Author Email Address

mailto: andrew.jeyaraj@mail.concordia.ca

## 7. Acknowledgements

The authors acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), Grant Number CRDPJ 538897-19 and RGPIN/5515-2019, and the Consortium de recherche et d'innovation en aérospatiale au Québec (CRIAQ). The presented work also benefited from the authors' collaboration within the AGILE4.0 project, which receives funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 815122.

The authors (1&3) would like to acknowledge the contributions of Alvaro Tamayo from Bombardier Aerospace for his expert insight through discussions, reviews, and suggestions.

## 8. Copyright Statement

## References


[1] Liscouët-Hanke S., "A Model Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures," Ph.D. Thesis, Université Toulouse III - Paul Sabatier, 2008.

[2] Liscouët-Hanke S., Maré J. C., and Pufe S., "Simulation framework for aircraft power system architecting," *J. Aircr.*, vol. 46, no. 4, pp. 1375–1380, Jul. 2009.

[3] Chakraborty I. and Mavris D. N., "Integrated Assessment of Aircraft and Novel Subsystems Architectures in Early Design," *J. Aircr.*, vol. 54, no. 4, pp. 1268–1282, 2017.

[4] Lammering T., "Integration of aircraft systems into conceptual design synthesis,"Ph.D. Thesis, RTWH Aachen, Aachen, 2014.

[5] Fioriti M., Boggero L., Tomasella F., Mirzoyan A., Isyanov A., and Prakasha P. S., "Propulsion and on-board system integration for regional, medium and long range jet with different level of systems electrification in the agile project," in *AIAA Joint Propulsion Conference*, 2018, AIAA 2018-4847.

[6] Boggero L., Fioriti M., Corpino S., and Ciampa P. D., "On-board systems preliminary sizing in an overall aircraft design environment," in *17th AIAA Aviation Technology, Integration, and Operations Conference, 2017*, AIAA 2017-3065.

[7] Boggero L., Ciampa P. D., and Nagel B., "An MBSE Architectural Framework for the Agile Definition of Complex System Architectures," in *AIAA Aviation Forum*, 2022, AIAA 2022-3720.

[8] Bussemaker J. H., Boggero L., and Ciampa P. D., "From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO," in *INCOSE International Symposium*, 2022.

[9] Bauer C., Lagadec K., Bès C., and Mongeau M., "Flight control system architecture optimization for fly-by-wire airliners," *J. Guid. Control. Dyn.*, vol. 30, no. 4, pp. 1023–1029, 2007.

[10] Zeidner L. E., Reeve H. M., Khire R., and Becz S., "Architectural enumeration & evaluation for identification of low-complexity systems," in *10th AIAA Aviation Technology, Integration and Operations Conference 2010, ATIO 2010*, vol. 3, AIAA 2010-9264.

[11] Becz S., Pinto A., Zeidner L. E., Khire R., Banaszuk A., and Reeve H. M., "Design system for managing complexity in aerospace systems," *10th AIAA Aviat. Technol. Integr. Oper. Conf. 2010, ATIO 2010*, vol. 2, 2010, AIAA 2010-9223.

[12] Garriga, A. G., Govindaraju P., Ponnusamy S. S., Cimmino N., and Mainini L., "A modelling framework to support power architecture trade-off studies for More-Electric Aircraft," *Transp. Res. Procedia*, vol. 29, pp. 146–156, Jan. 2018.

[13] Garriga A. G., Mainini L., and Ponnusamy S. S., "A machine learning enabled multi-fidelity platform for the integrated design of aircraft systems," *J. Mech. Des. Trans. ASME*, vol. 141, no. 12, Dec. 2019.

[14] Judt D. and Lawson C. P., "Methodology for automated aircraft systems architecture enumeration and analysis," in *AIAA ATIO*, 2012, AIAA 2012-5648.

[15] Judt D. M. and Lawson C., "Development of an automated aircraft subsystem architecture generation and analysis tool," *Eng. Comput.*, vol. 33, no. 5, pp. 1327–1352, Jul. 2016.

[16] Chakraborty I. and Mavris D. N., "Heuristic definition, evaluation, and impact decomposition of aircraft subsystem architectures," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, AIAA 2016-3144.

[17] Jeyaraj A. K., Tabesh N., and Liscouet-Hanke S., "Connecting Model-based Systems Engineering and Multidisciplinary Design Analysis and Optimization for Aircraft Systems Architecting," in *AIAA AVIATION FORUM*, 2021, AIAA 2021-307.

[18] Bussemaker J. H., Ciampa P. D., and Nagel B., "System architecture design space exploration: An approach to modeling and optimization," in *AIAA Aviation Forum*, 2020, vol. 1 PartF, pp. 1–22, AIAA 2020-3172.

[19] SAE International, "ARP4761:Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.," 1996.

[20] Bussemaker J. H., Ciampa P. D., and Nagel B., "System Architecture Design Space Modeling and Optimization Elements," in *32nd Congress of the International Council of the Aeronautical Sciences*, 2021, ICAS 2021-10.3.1.

[21] Bussemaker J. H. and Boggero L., "Technologies for Enabling System Architecture Optimization," in *ODAS Symposium*, 2022.

[22] Leveson N., Wilkinson C., Fleming H. C., Thomas J., and Tracy I., "A Comparison of STPA and the ARP 4761 Safety Assessment Process 1," 2014.

[23] Blank J. and Deb K., "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.