

Article

Evaluating MBSE Approaches and Tools for Aircraft Design and Certification: A Comparative Perspective

Claudio Mirabella * , Michele Tuccillo  and Pierluigi Della Vecchia * 

Department of Industrial Engineering, University of Naples Federico II, Via Claudio 21, Palazzina 11, I piano, 80125 Naples, Italy; michele.tuccillo@unina.it

* Correspondence: claudio.mirabella@unina.it (C.M.); pierluigi.dellavecchia@unina.it (P.D.V.); Tel.: +39-338-471-3099 (C.M.)

Abstract

This article evaluates two model-based systems engineering (MBSE) toolchains that support aircraft certification under EASA CS-23 Amendment 6. Airworthiness requirements and associated acceptable means of compliance are digitalized as Systems Modeling Language (SysML) models that preserve document structure and encode parameters and expressions needed for substantiation. The maneuvering and gust flight envelope required by CS-23 Subpart C is used as a representative case to compare workflow integration, robustness, and artifact generation. One implementation combines Eclipse Papyrus with MATLAB to export and parse the SysML model and to execute automated calculations and reporting. The second uses CATIA Magic Systems of Systems Architect (MSoSA) to export stereotype fields to JSON and to run C++ routines orchestrated by activity diagrams. Both toolchains generate certification-relevant outputs, including design airspeeds, limit load factors, and flight envelope plots, while improving traceability relative to document-centric practice. The comparison indicates that the Papyrus/MATLAB approach supports rapid prototyping but is more sensitive to regulatory text changes, whereas the MSoSA-based approach reduces dependence on text–pattern parsing and provides more integrated execution. These results suggest that MBSE can improve the efficiency of preparing certification evidence, with adoption trade-offs driven by licensing cost, integration effort, and organizational maturity.

Keywords: MBSE; SysML; general aviation; aircraft certification; digital certification

1. Introduction

Certification is a systematic, mandatory process applied to complex industrial products to verify conformity with regulatory, safety, and performance standards, particularly where operations pose a high risk to property and human life. Aerospace and aeronautical products are subject to stringent scrutiny by regulatory agencies such as the Federal Aviation Administration (FAA) and the European Union Aviation Safety Agency (EASA). This oversight has significant implications for design and manufacturing organizations, which must not only engage effectively with regulatory authorities but also establish internal procedures to produce, maintain, and issue the data and artifacts required to demonstrate compliance with all applicable requirements. Quantifying the impacts associated with this process has proved difficult and cumbersome [1].

The FAA and EASA have adopted a set of documents issued by the American Society for Testing and Materials (ASTM) International to promote regulatory interoperability between North American and European jurisdictions [2]. While this policy offers significant



Academic Editor: Tugba Karabiyik

Received: 16 March 2026

Revised: 21 April 2026

Accepted: 27 April 2026

Published: 29 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

benefits—such as flexibility in selecting certification bases—it also introduces challenges, including the complexity of cross-referencing multiple documents and the risk of human error in manual processes [3–5]. Moreover, the Consensus Standard comprises numerous standards, and applicants are often unable to determine in advance which set may be sufficient or even applicable to the design they propose [6]. Although experts and certification engineers can derive certification-constrained approaches [7,8]—addressing different aspects of the certification process—design and manufacturing organizations operate in highly competitive environments, where responsiveness may be more valuable than technological achievements.

Aviation regulators are actively convening conferences and workshops and engaging industry stakeholders to shape the future certification landscape. For example, the EASA Part 21 Workshop and Certification Conference 2026 [9] promoted the strategic objective of a fully digital, data-centered certification process by 2030, guided by the following principles:

- Sustainability of the digital certification process, with emphasis on standardization and cost efficiency.
- Security and safety, with focus on secure digital access, data protection, and safe deployment of products and services.
- Compliance, addressing data management, data interoperability, and standardization of data formats and rules.
- Scalability, with the goal of reducing costs by broadening the applicability of tools and techniques, including potential transfer to other industrial sectors.
- Integration, with the objective of enabling rapid and seamless deployment of the proposed techniques.

Although the proposed timeline is ambitious, it underscores the importance of investigating digital tools and methods that support the certification process.

The introduction of new technologies is hindered by the lack of precise procedures, requirements, and safety standards comparable to those applied to conventional aircraft architectures. A notable example is the evaluation of equivalent safety and handling qualities under emergency conditions—specifically, with one or more engines inoperative—for aircraft employing distributed-propulsion mechanisms [10,11]. This may lead to a decrease in competitiveness and innovation over time in the aeronautical industry, especially in the general aviation market [12]. EASA emphasizes that certification is largely an engineering process in which applicants demonstrate compliance through certification documents. The methodologies discussed in this work aim to establish an MBSE toolchain that supports the integrated management of these documents, improving traceability and maintainability within a certification-item database.

To pursue this goal, the University of Naples Federico II (UNINA), in collaboration with Politecnico di Milano (POLIMI) and Politecnico di Torino (POLITO), launched the Digital Innovative General Aviation Aircraft Certification (DIGACE) research project. This initiative applies Model-Based Systems Engineering (MBSE) techniques to digitalize airworthiness rules and Means of Compliance (MoC), integrating content from EASA CS-23, Amendment 6, and ASTM standards [13,14]. More broadly, MBSE approaches have been applied to complex industrial products to link stakeholder needs and operational capabilities to system architectures and design features [15]. Other industrial sectors with comparable complexity, and with regulatory involvement across design and manufacturing phases, have also adopted MBSE techniques to improve communication and to support efficient retrieval and delivery of certification-relevant information [16].

The adoption of MBSE has led to the development of specialized tools and encoding mechanisms, including the Systems Modeling Language (SysML). SysML provides a formal, structured notation for representing complex systems and managing certification-relevant

information. The digital certification framework developed in this work uses SysML together with MATLAB/Simulink and C++ code suites and is demonstrated through two implementations: one based on Eclipse Papyrus and another using CATIA Magic Systems of Systems Architect (MSoSA). These implementations support automated parsing, computation, and generation of certification artifacts, thereby streamlining key certification activities for manufacturers and potentially increasing the efficiency of data and information exchange with the regulatory agencies during the certification process. More broadly, MBSE methodologies can improve certification efficiency [17–19], support the integration of innovative technologies [20,21], and help preserve safety margins for unconventional designs [22–24], at least at the certification level. The certification process is based on a complex interchange of information under the form of certification documents and artifacts between the applicants and the regulators, an application possibly ideal for the introduction of MBSE techniques [25]. In addition, the adoption of MBSE in industrial contexts is supported on economic grounds [26], particularly for complex products with an intricate product life cycle, using holistic and comprehensive MBSE approaches [27,28]. Other digitalization approaches have been addressed in the literature [29–32], and there is a stimulating debate in the aviation research and industrial community.

This study examines EASA CS-23 Subpart C, which specifies structural requirements for aircraft certification. When applying for a new Type Certificate (TC), applicants must demonstrate that the design provides adequate structural strength for its intended operational use [33].

Regulatory guidelines issued by EASA and FAA prescribe the construction of a flight envelope that combines maneuvering and gust loads. The resulting load factors are used to size critical structural components, which are subsequently verified during ground testing. This phase establishes the limit loads, defined as the maximum loads the airframe can withstand without permanent deformation [34].

The flight envelope is therefore a fundamental element in certification, ensuring compliance with safety standards and structural integrity under operational conditions. Its accurate determination is essential for both conventional and innovative aircraft configurations, as it directly influences design decisions and certification outcomes. For these reasons, the maneuvering and gust flight envelope is used in this study as a representative example to demonstrate the capabilities of the proposed MBSE-based digital certification toolchains.

This study has the following structure:

- Section 1 (this section) introduces the study, outlines its motivation, and briefly describes the challenges of aeronautical certification in the modern aviation industry, with emphasis on enabling new technologies while preserving the value of the aeronautical product.
- Section 2 describes the EASA certification framework relevant to this work and clarifies that the scope is limited to CS-23 Subpart C (Structure), together with the corresponding acceptable means of compliance used for substantiation.
- Section 3 presents the reference aircraft architecture used to illustrate the certification and substantiation calculations, including the minimum required input data and the resulting airspeeds and limit load factors derived from maneuvering and gust flight-envelope calculations.
- Section 4 describes the conceptual structure and workflow of the digital certification framework.
- Section 5 compares the two implementations presented in this study:
 - Section 5.1 describes the Eclipse Papyrus/MATLAB (R2024b, Update 8, 24.2.0.3157250) digital certification framework, including its structure and the data it processes.

- Section 5.2 describes the CATIA MSoSA/C++ digital certification framework, including the data exported from the model and provided to the C++ substantiation routines.
- Section 6 presents selected outputs generated by the digital certification frameworks:
 - Section 6.1 describes the digital database generated from the SysML model and the certification artifacts produced automatically by the toolchain.
 - Section 6.2 describes the CATIA MSoSA execution workflow and the resulting data products and certification artifacts.
- Section 7 defines the KPIs used to assess the two MBSE methodologies, discusses the results, and provides a comparative evaluation.
- Section 8 concludes the study and discusses the findings in the context of a broader shift from document-based to model-based certification processes.

2. Certification Landscape

This section examines the *European Union Aviation Safety Agency* (EASA) Certification Specifications for general aviation aircraft (CS-23), Amendment 6, with particular attention to Subpart C (Structure). Because these specifications state requirements at a high level, they rely on external documents for compliance, notably the consensus standards issued by ASTM International (formerly the *American Society for Testing and Materials*). Under each CS-23 paragraph, the relevant ASTM standard is explicitly identified as the reference *Acceptable Means of Compliance* (AMC). The CS-23 framework also permits the applicant to use two alternative AMCs: EASA CS-23, Amendment 4—which is scheduled for discontinuation and is currently accepted only for aircraft to be certified in the European Union—and the Certification Specifications for *Very Light Airplanes* (CS-VLA). The latter is narrowly scoped and applies only to aircraft with a Maximum Takeoff Weight (MTOW) of 450.00 kg or less and a maximum number of passengers not exceeding 2. This work references AMC1—namely, the ASTM International standards—to develop a digital certification framework. Two approaches for establishing this framework are described in the following sections. Figure 1 summarizes the current EASA certification landscape and explicitly references the **ASTM F3116/F3116M Standard Specification for Design Loads and Conditions** [35]. Other ASTM standards that address the high-level requirements of Subpart C include:

- **F3396/F3396M Standard Practice for Aircraft Simplified Loads Criteria** [36], which focuses on small, simple, and very light airplanes with a stalling speed not exceeding 45.00 knot.
- **F3331-18 Standard Practice for Aircraft Water Loads** [37], which focuses on structural design loads for seaplanes.

ASTM publishes additional standards that, in some sections, address structural design loads and conditions relevant to Subpart C, such as **F2245-23 Standard Specification for Design and Performance of a Light Sport Airplane** [38]. Although F2245-23 also addresses other EASA Subparts and remains applicable to Level 1 aircraft, this work focuses on F3116/F3116M because it applies to all aircraft eligible for certification in the Normal category. These standards provide detailed procedures for constructing the maneuvering and gust flight envelope, as required by aeronautical regulators, including the functional relationship between airspeed V and the applicable limit load factors n . The applicant must demonstrate compliance with strength requirements for all relevant combinations of airspeed and limit load factor within the resulting flight envelope. The ASTM standard specifies limitations for both the maneuvering limit load factor and the gust limit load factor. For example, the maneuvering limit load factor is defined as

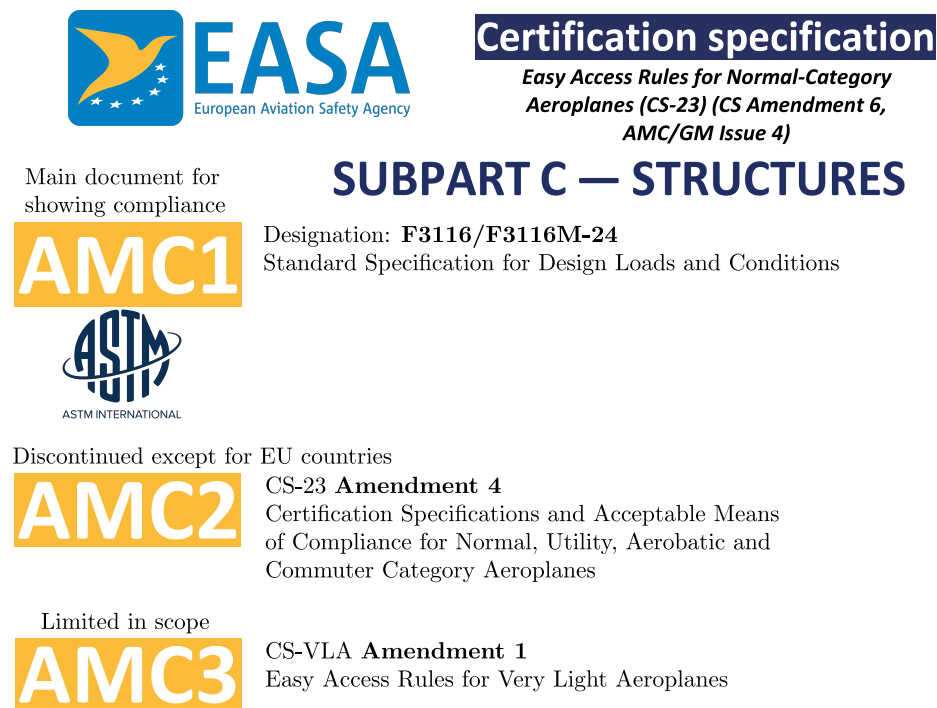


Figure 1. The reference certification landscape for the work presented in this article. The EASA CS-23, Subpart C, refers to the ASTM F3116/F3116M Standard for substantiating the high-level requirements and showing compliance.

$$n_m = 2.1 + \frac{24,000}{W + 10,000} \quad \text{with } W \text{ in [lb]} \quad (1)$$

with an upper limit of 3.8 for airplanes not approved for aerobatics and 6.0 for airplanes approved for aerobatics. The negative maneuvering limit load factor must not be less than $0.4 \cdot n_m$ for airplanes not approved for aerobatics and $0.5 \cdot n_m$ for airplanes approved for aerobatics. The two methodologies proposed in this work enable the user to apply relations such as Equation (1) automatically, ensuring that boundaries are enforced and checked against appropriate limits. Therefore, the calculations and representation of the maneuvering and gust flight envelope can be considered a standard test case for the methodologies proposed in this work, capable of clarifying their applicability in the broader context of general aviation certification.

3. Reference Aircraft Architecture

Figure 2 shows the reference aircraft architecture used to generate the certification artifacts, including the maneuvering and gust flight envelope. The aircraft is loosely based on the Tecnam P2012 Traveller and is representative of a Level 4, low-speed commuter-category design. Table 1 summarizes the input data required to initiate the flight-envelope calculations. As a preliminary step, the gust intensities are determined automatically from the applicable certification requirements. For brevity, several additional preliminary steps analogous to those described in this section are omitted, such as the determination of the maximum horizontal flight speed at maximum continuous power.

Tables 2 and 3 report the computed airspeeds and corresponding limit load factors obtained with the Eclipse Papyrus/MATLAB and CATIA MSoSA/C++ digital certification frameworks. Because the two runs use slightly different computational strategies and input values, the resulting outputs differ accordingly. These results indicate that both implementations can accommodate variations in the aircraft input dataset and can therefore be applied to different aircraft architectures. The main objective of this work is to evaluate and

compare two different automated approaches for generating these certification artifacts, so there is no need for exact correspondence in numerical results. However, the two solutions are in close agreement.

Table 1. Reference aircraft input data for the evaluation of the maneuvering and gust flight envelope.

Input Data	Value
Maximum lift coefficient, $C_{L_{\max}}$	1.46
Minimum lift coefficient, $C_{L_{\min}}$	−0.80
Lift curve slope coefficient, $C_{L_{\alpha}}$	4.71 rad ^{−1}
Wing Loading (SI units), $\frac{W}{S}$	1352.79 kgm ^{−2}
Wing Loading (Engineering units), $\frac{W}{S}$	28.25 lb ft ^{−2}
Density at the reference altitude, ρ_{ref}	0.85 kg m ^{−3}
Density at sea level, ρ_0	1.22 kg m ^{−3}
Mean aerodynamic chord, MAC	1.82 m
Aircraft mass (Engineering units), M	8113.00 lb
Gust intensity at cruise speed, U_{deC}	15.24 m s ^{−1}
Gust intensity at dive speed, U_{deD}	7.62 m s ^{−1}
Gust intensity for rough air speed, U_{deB}	20.12 m s ^{−1}
Max. hor. airspeed at sea level, V_H	93.38 m s ^{−1}
Max. hor. airspeed at reference alt., V_H	104.65 m s ^{−1}
Reference altitude h_{ref}	12,000.00 ft
Max. cont. power (Eng. units) P_{MC}	375.00 hp × 2
Max. cont. power (SI units) P_{MC}	279.64 kW × 2

Table 2. A summary of the geometrical analysis performed by the Papyrus/MATLAB architecture during the maneuvering and gust flight envelope evaluation procedure.

	PARAMETERS	h = Sea Level	h = 12,000.00 ft
Airspeeds	Positive stall speed, $V_{S_{\text{clean}}}^+$	38.89 m s ^{−1}	46.72 m s ^{−1}
	Negative stall speed, $V_{S_{\text{clean}}}^-$	52.54 m s ^{−1}	63.11 m s ^{−1}
	Max. hor. design speed, V_H	93.38 m s ^{−1}	104.65 m s ^{−1}
	Design cruising speed, V_C	84.04 m s ^{−1}	89.00 m s ^{−1}
	Design maneuvering speed, V_A	71.98 m s ^{−1}	86.46 m s ^{−1}
	Negative design maneuvering speed, V_G	61.50 m s ^{−1}	73.87 m s ^{−1}
	Design dive speed, V_D	117.22 m s ^{−1}	124.14 m s ^{−1}
	Design rough air speed, V_B	84.04 m s ^{−1}	89.00 m s ^{−1}
Limit Load Factors	Stalling limit load factor, n_S	±1.00	±1.00
	Cruising limit load factor, n_C	3.43	3.43
	Maneuvering limit load factor, n_A	3.43	3.43
	Dive limit load factor, n_D	3.43	3.43
	Neg. maneuvering limit load factor, n_G	−1.37	−1.37
	Neg. cruising limit load factor, n_F	−1.37	−1.37
	Neg. dive limit load factor, n_E	3.43	3.43

In [13], the MATLAB-based digital certification framework is extended to additional CS-23 Subparts. In future work, the authors will apply the CATIA MSoSA digital certification framework to three conventional aircraft architectures and generate the certification artifacts required to substantiate structural compliance, supported by a Python (2.7 for internal scripting, 2.9 for external tools) companion library. The present study is limited in scope and provides an overview of two MBSE approaches to aeronautical certification in general aviation. The analysis is restricted to CS-23 Subpart C acceptable means of compliance and focuses specifically on the derivation of the maneuvering and gust flight

envelope for aircraft eligible for certification under Part 23. The procedure is broadly applicable to general aviation aircraft.

Table 3. A summary of the geometrical analysis performed by the CATIA MSoSA/C++ architecture during the maneuvering and gust flight envelope evaluation procedure.

	PARAMETERS	h = Sea Level
Airspeeds	Positive stall speed, $V_{S_{clean}}^+$	39.42 m s ⁻¹
	Negative stall speed, $V_{S_{clean}}^-$	53.26 m s ⁻¹
	Max. hor. design speed, V_H	98.31 m s ⁻¹
	Design cruising speed, V_C	84.68 m s ⁻¹
	Design maneuvering speed, V_A	73.10 m s ⁻¹
	Negative design maneuvering speed, V_G	62.46 m s ⁻¹
	Design dive speed, V_D	117.02 m s ⁻¹
Limit Load Factors	Stalling limit load factor, n_S	±1.00
	Cruising limit load factor, n_C	3.44
	Maneuvering limit load factor, n_A	3.44
	Dive limit load factor, n_D	3.44
	Neg. maneuvering limit load factor, n_G	−1.38
	Neg. cruising limit load factor, n_F	−1.38
	Neg. dive limit load factor, n_E	3.44

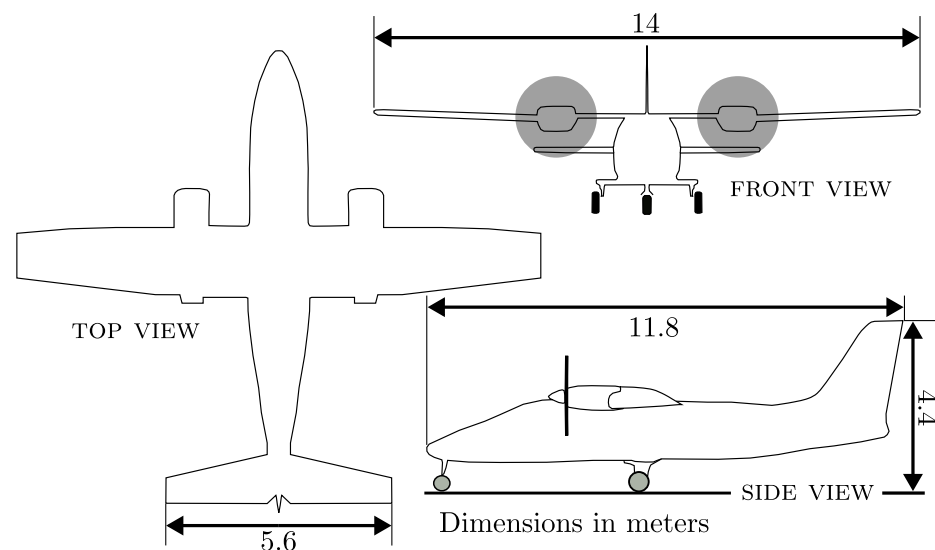


Figure 2. Three-view representation of the reference aircraft architecture analyzed using the two digital certification frameworks.

4. The Digital Certification Workflow

Figure 3 presents the general conceptual design of the digital certification framework. The general workflow that an applicant should follow when using the MBSE digital certification framework is as follows:

1. First, the applicant encodes the information required by the framework in a usable format. Certification and substantiation rules are encoded by constructing a SysML model. Regardless of the tool used, the applicant must digitally formalize the regulations, defining an appropriate set of stereotypes with properties that can store data, formulas, and other relevant information, building a usable SysML model in a one-to-one correspondence with the original documents. In parallel, the applicant must build a digital database containing a substantial amount of data describing the design

architecture under consideration. The geometry and, at a minimum, the wing–body aerodynamic characteristics must be included in a consistent input file.

2. Once the inputs are prepared, the framework parses the model and builds the structured variables required to inform the calculation routines, whether the applicant uses the Papyrus/MATLAB or the CATIA MSoSA/C++ implementation proposed in this article.
3. The routines are then populated automatically with regulatory data and formulas. The calculations begin, and the results are checked against the regulatory boundaries.
4. Finally, the calculation routines produce data and certification information that can be used to create certification artifacts—figures, tables, and entire certification documents in Word or PDF formats. These artifacts remain necessary to demonstrate compliance to aeronautical regulators, in accordance with established procedures used across the aviation industry.

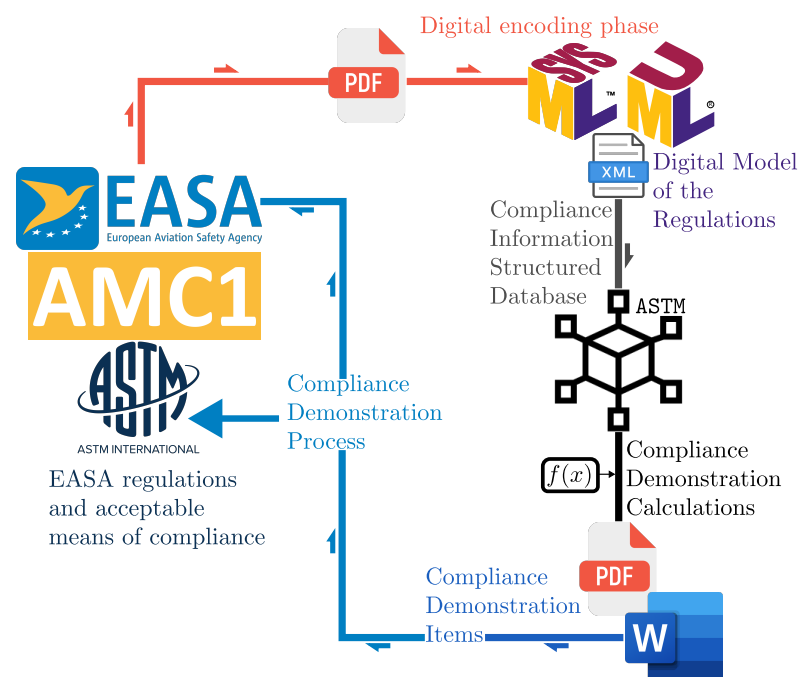


Figure 3. The digital certification framework and its general conceptual design.

Overall, the generation of certification artifacts is consistent and coherent and is managed automatically through a stable, well-defined process. The certification calculations underlying these artifacts are fully automated, which minimizes the risk of errors during this phase. The digital certification framework has the following disadvantages:

- Although its structure and logic are relatively straightforward, the framework’s functionality depends on the regulations and is highly sensitive to changes in certification documents. Updating the SysML model can be burdensome and requires trained personnel. Even when the necessary expertise is available within an organization, implementing changes to the SysML model requires time and financial resources that are often limited in small manufacturing companies in the general aviation sector. A higher level of resilience to this defect has been included in the second digital certification framework proposed in this work, as illustrated in Section 5.2.

- The aircraft input data may require a level of detail that is not available at all stages of the certification and manufacturing process. As a result, the framework must remain limited in scope and focus on specific subparts. This constraint can be viewed as a limitation; however, it can also improve maintainability because it aligns with guidance from the *International Council on Systems Engineering* (INCOSE), which recommends avoiding an overly broad SysML model that could compromise maintainability [39].
- Finally, although the framework illustrates the application of MBSE methods to a complex industrial process, it ultimately relies on document-based outputs to demonstrate compliance with aeronautical regulations. The certification artifacts generated at the end of execution remain documents, such as substantiation reports, tables, and graphs.

The complexity of the SysML model lies primarily in defining and implementing an appropriate set of stereotypes to capture the properties of the modeled objects. For the EASA regulations and the associated acceptable means of compliance, the work presented in this article relies mainly on two possible choice for user-defined properties:

- A property defined as a zero-to-many multiplicity vector of strings. The purpose of this field is to capture formulas, equations, disequations, and other complex information associated with the model's elements.
- A property defined as a zero-to-many multiplicity vector of float numbers. The purpose of this field is to capture simple numbers and numerical data included in the text of ASTM documents, paragraphs, and subparagraphs.

The user-defined stereotypes will capture the EASA CS-23 paragraphs and the ASMT International F3116/F3116M paragraphs and subparagraphs. Once these SysML constructs are defined, the digital certification process can be established, either through dedicated export-and-parse workflows or through proprietary tools available within the modeling environment. Figure 4 shows an example of a SysML Profile Diagram in the Eclipse Papyrus modeling environment, where users define new stereotypes. In this example, a new SysML element is defined that inherits the properties of the SysML *Requirement* element. Several properties are also introduced:

- *TestValue1* and *TestValue2* are string properties with zero-to-many multiplicity. These strings may contain any admissible values, including formulas and data.
- *CertificationLevel* denotes the EASA certification level, from Level 1 (aircraft with a maximum of two passengers) to Level 4 (aircraft with a maximum of nineteen passengers).
- *PerformanceLevel* can take the values *high speed* or *low speed* and identifies the aircraft performance level.

The stereotypes used in the models discussed in this work are derived from the example shown in Figure 4, with minor refinements that improve model flexibility and usability.

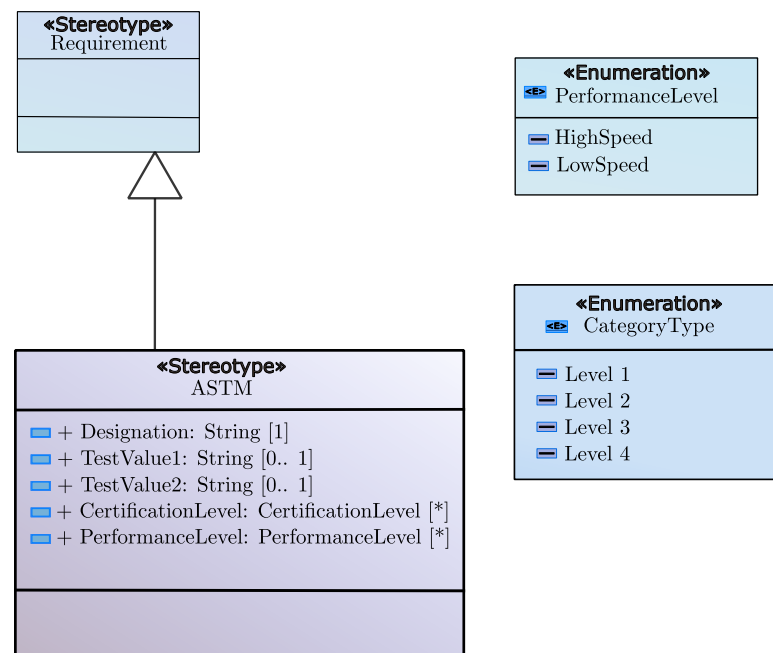


Figure 4. An example of a Profile Diagram. The stereotype's properties show the multiplicity symbols indicating the lower and upper bound of the the multiplicity interval, such as [0..1] and [*], where the asterisk indicates an unlimited multiplicity. Adapted from [13] for illustrative purposes.

5. Comparison of the Two Methodologies

This section compares two strategies for implementing MBSE techniques in the context of general aviation aircraft certification. The scope of the comparison is limited to Subpart C.

- Section 5.1 describes the Eclipse Papyrus modeling approach and the associated MATLAB/Simulink companion code, which parses the resulting SysML model and initiates certification-related calculations. This implementation serves as an effective research tool for evaluating a broader digital certification framework. However, the reliance on Regular Expressions during the parsing phase is deemed inadequate for industry-level context.
- Section 5.2 presents an alternative implementation using CATIA MSoSA features, including Activity Diagrams for automated execution of external code. This approach offers improved capability to capture information from ASTM standards through user-defined stereotypes and enhances reliability by eliminating the use of Regular Expressions in parsing.

5.1. DIGACE Implementation

Figure 5 illustrates the structure of the Papyrus/MATLAB digital certification framework by tracing its execution logic and summarizing its functional architecture. The DIGACE implementation consists of three phases designed to digitalize certification specifications compliance verification. MATLAB's expressive syntax supports class definitions whose methods can be generated directly from information parsed from the SysML model by processing the corresponding UML file. The resulting MATLAB suite operates without invoking Simulink routines or advanced MATLAB toolboxes. Instead, it provides a focused computational framework that automatically constructs the maneuvering and gust flight envelope using the information encoded in the SysML model. Although the parsing strategy adopted here is simple and effective, it can be sensitive to changes in the substantiation documentation. The ASTM consensus standards are updated and amended regularly, and revisions between successive versions can be substantial.

Figure 6 provides an overview of the Eclipse Papyrus modeling environment and its representation of the SysML hierarchy. The paragraph and subparagraph structure of the source documents is modeled with a one-to-one correspondence, preserving the logical organization of the certification information. In this implementation, the parser relies on regular expressions and navigates the hierarchy using model-element identifiers. Once the target element is located, the parser scans its textual content using the regular expressions associated with that specific requirement. However, this coupling between text and regular expressions can be disrupted when revised versions of the certification documents are issued, leading to parsing errors. Although the parser is reliable for a fixed version of the model, it is less robust when applied across multiple document amendments, which limits its suitability for a general digital certification workflow. In this implementation, the SysML model provides fewer automation and integration features; nevertheless, it remains capable of capturing the certification information and encoding it in a usable textual form. The automation can be implemented using external tools, such as MATLAB.

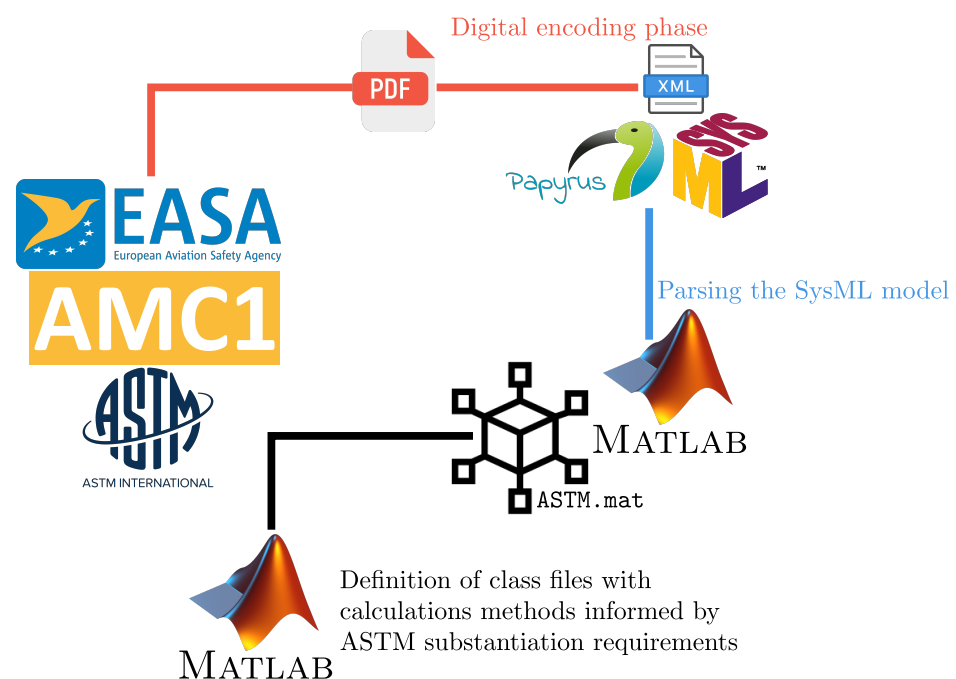


Figure 5. The DIGACE research project implemented a digital certification framework built upon the Eclipse Papyrus SysML modeling environment. The SysML model automatically generates a UML file (which can be considered similar to a complex XML file), which can be parsed using MATLAB. The parsing strategy is based on regular expressions. Although this implementation is less reliable, the process of constructing the facilities is relatively quick and has a simple underlying logic.

The execution of this implementation of the digital certification framework has the following phases.

Phase 1: Modeling. Certification data are encoded using the structured syntax of SysML into a digital model, producing a UML file compliant with the International Council on Systems Engineering (INCOSE) standards. This file, which resembles a conventional XML file in its structure, contains textual information extracted from EASA CS-23 and Means of Compliance (MoC) documents, along with metadata describing relationships, stereotypes, and model elements. The output of this phase is a structured variable containing all the certification requirements and the substantiation methods, ready to inform MATLAB functions implemented in dedicated MATLAB classes.

Phase 2: Parsing and Computation. A MATLAB program reads the UML file and parses certification-related information using regular expressions. Regular expressions are relatively simple to use and manipulate, although the resulting parsing procedure could be considered insufficiently reliable for industry-rated applications. The extracted data populate structured variables that define limits, constraints, and parameters for Subpart C calculations. For example, flight envelope construction references ASTM F3116/F3116M *Standard Specification for Design Loads and Conditions*, Section 4.4, *Flight Loads* [35], p. 3.

Phase 3: Artifacts Generation. Parsed data and computational results are used to generate flight envelope diagrams and formal certification reports through MATLAB's *Report Generator* API. These outputs form certification artifacts organized into three categories: Certification Demonstration Data, Items, and Artifacts. Regulatory compliance is automatically ensured and systematically tracked using a Certification Compliance Checklist, a feature of the process that increases traceability.

- ▼ F3116/F3116M-23a Standard Specification for Design Loads and Conditions
 - ▼ F3116/F3116M-23a Standard Specification for Design Loads and Conditions
 - > «ASTM» 3. Terminology
 - ▼ «ASTM» 4. Flight Loads
 - > «ASTM» 4.1 Flight Loads
 - > «ASTM» 4.2 Flight Loads
 - > «ASTM» 4.3 Flight Loads
 - > «ASTM» 4.4 Flight Loads
 - ▼ «ASTM» 4.5 Flight Loads
 - ▼ «ASTM» 4.5.1 Flight Loads
 - «ASTM» 4.5.1.1 Flight Loads
 - «ASTM» 4.5.1.2 Flight Loads

«ASTM» 4.5.1.1 Flight Loads		
SysML 1.6	Id	4.5.1.1 Flight Loads
UML	Name	
Text		2.1+(24000/(W+10000))
Comments		where
Profile		W = design maximum takeoff weight (lb), except that n need not be more than 3.8
Advanced		
Is abstract	☐ true ☒ false	Is leaf

2.1+(24000/(W+10000))

where

W = design maximum takeoff weight (lb), except that n need not be more than 3.8

Figure 6. The Eclipse Papyrus containment tree and the ASTM F3116/F3116M 4.5.1.1 Limit Maneuvering Load Factor paragraph. An enlarged perspective on the textual content of the model's element is given.

The MATLAB programming environment provides extensive capabilities for generating certification reports efficiently and with a high degree of automation. In this work, the implementation combines MATLAB built-in functions with custom class files, enabling a relatively short development cycle. By contrast, the Eclipse Papyrus modeling environment provides fewer automation features than CATIA MSoSA for orchestrating workflow execution, which results in a less integrated SysML infrastructure for the proposed certification process. Consequently, the resulting digital certification infrastructure provides

more limited support for traceability and for the verification and validation of the generated certification artifacts, although the time and cost required for implementation remain relatively low. Moreover, the user must rely on a set of external tools and workflows that operate outside the SysML modeling environment, which makes the end-to-end process less seamless in practical use. This methodology enables automated extraction, calculation, and reporting, reducing manual effort and improving accuracy while maintaining compliance with regulatory standards.

5.2. CATIA MSoSA Implementation

Figure 7 illustrates how the CATIA/C++ implementation differs from the preceding workflow: both the SysML-based digital encoding and the parsing procedure are performed within a single environment, as indicated by the blue square enclosing CATIA Magic Systems of Systems Architect (MSoSA). The CATIA MSoSA implementation was evaluated to determine its capability to provide a more integrated approach to MBSE-based certification. Unlike the previous workflow, this environment enables direct execution of tasks within the modeling platform through Activity Diagrams, reducing reliance on external tools. In this approach, parsing is fully decoupled from the definition of the calculation functions and is self-contained within the CATIA MSoSA environment. The workflow is expected to be robust because it leverages the properties of the stereotypes: when an element has the user-defined stereotypes and its fields are non-empty, their contents are exported to a JSON file in a consistent structure that can be passed to the C++ routines. This separation improves the maintainability and usability of the MBSE-based digital certification framework.

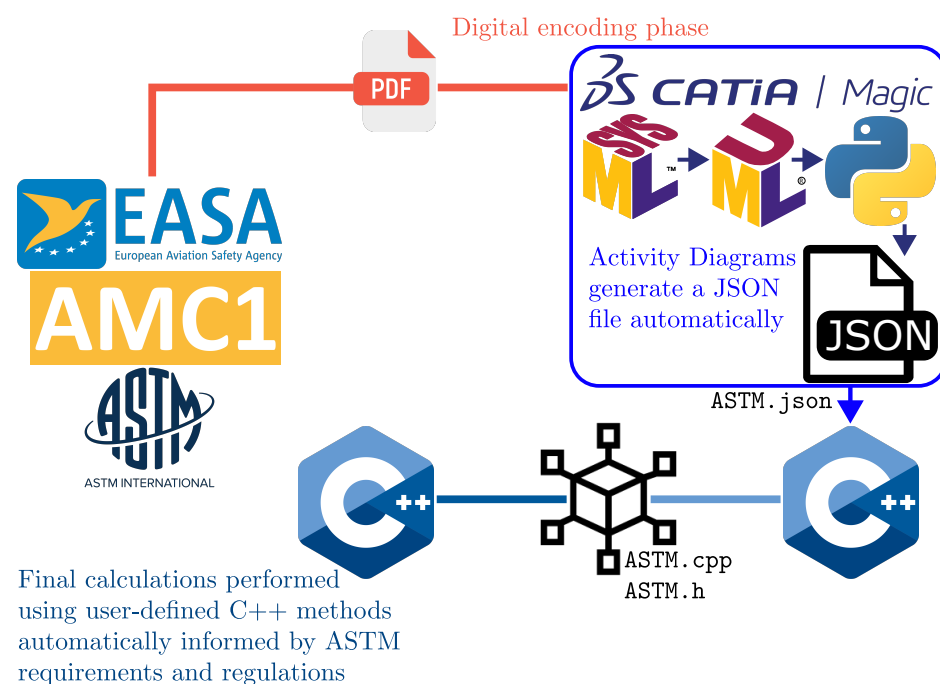


Figure 7. The CATIA MSoSA digital certification framework relied upon the definition of internal Activity Diagrams. A companion, internally defined Python code parses the model and saves the ASTM substantiation requirements into a JSON file. The JSON file is then passed to a C++ code that creates class files with calculation methodologies directly informed by ASTM regulations.

Figure 8 presents an example of an Activity Diagram, as shown in the CATIA MSoSA modeling environment. This particular example refers to the definition and execution of an internal Python routine that scans the model and parses its elements into a JSON file, ready to be used by the C++ routines.

Figure 9 shows the user-defined SysML stereotype used to represent the paragraph and subparagraph structure of the ASTM International standards. Although the stereotype is simple in its definition, it captures most of the information required to substantiate compliance with the high-level EASA CS-23 requirements. The CS-23 requirements are represented using a conceptually similar stereotype with slightly different properties. Notably, these stereotypes retain the `TestValue` fields, defined as zero-to-many string vectors. Additional properties encode certification-relevant characteristics, including certification level, performance level, whether the applicant seeks approval for aerobatic maneuvers, and the installed engine type. The process begins by extracting data from the SysML model's XML structure to generate a JSON file. This file serves as input for a suite of standalone C++ routines designed to perform certification-related calculations, specifically for flight envelope determination in accordance with ASTM F3116/F3116M, Section 4.4, *Flight Loads* [35], p. 3—corresponding to AMC1 of EASA CS-23, Amendment 6. Data extraction is automated using a Python script triggered through Activity Diagrams, ensuring consistency regardless of model updates. The extraction follows the ASTM standard stereotype properties, taking advantage of the stereotype's fields. Please refer to Section 4 for a thorough description of the stereotypes and their properties.

It is easy to instruct the Python routine to search for the model's elements with non-empty fields. The resulting JSON file organizes all required parameters in a format compatible with the C++ suite, which then generates the flight envelope diagram. In this manner, the problems associated with the parsing procedure illustrated in Section 5.1 are circumvented efficiently, with an increase in execution and results stability.

Figure 10 presents the containment tree in the CATIA MSoSA SysML modeling environment. This view provides a convenient representation of the regulatory SysML model's hierarchical structure and preserves the paragraph and subparagraph organization of the source documents. This hierarchy can be leveraged by the parser, which can be configured to query model elements at specific levels of the structure. Figure 11 shows an example of an element tab in the CATIA MSoSA modeling environment. The tab displays the selected model element and its associated properties. The properties `TestValue1` and `TestValue2` are populated with string values extracted from the regulatory text. Once exported to the JSON file, these values can be used to parameterize the calculation methods and initiate the certification-calculation phase.

The C++ suite has the following structure:

- A C++ source file accesses the JSON file containing ASTM data and formulas and instantiates a structured object. This object supplies functions and methods in other classes with data and formula strings, which are then converted into reusable functions.
- A set of classes defines the flight-envelope boundaries and other preliminary quantities, such as standard-atmosphere properties at the reference altitude. The design airspeeds are computed, including the maximum design horizontal airspeed V_H .
- Additional classes implement the construction of the flight envelope and generate the output files used for the graphical representation of the maneuvering and gust flight envelope. The final PDF figure is produced with a series of L^AT_EX standalone scripts that can be executed manually or invoked automatically by the C++ code. The process automatically identifies the final maneuvering and gust flight-envelope geometry and performs checks along each branch.

This approach demonstrates the feasibility of integrating modeling, data extraction, and computation within a single platform. By orchestrating tasks through Activity Diagrams, CATIA MSoSA streamlines certification workflows, minimizes manual intervention, and ensures traceability of all operations. In practical use, the CATIA MSoSA environment

can be extended with custom libraries that are executed directly from appropriately defined Activity Diagrams, while the underlying data-exchange steps remain transparent to the end user. As a result, an aeronautical design organization can focus training on CATIA MSoSA, without requiring users to interact directly with the backend code that performs the substantiation calculations. In the application presented in this work, the substantiation routines were implemented in C++; however, the choice of programming language does not constitute a limitation of the proposed approach.

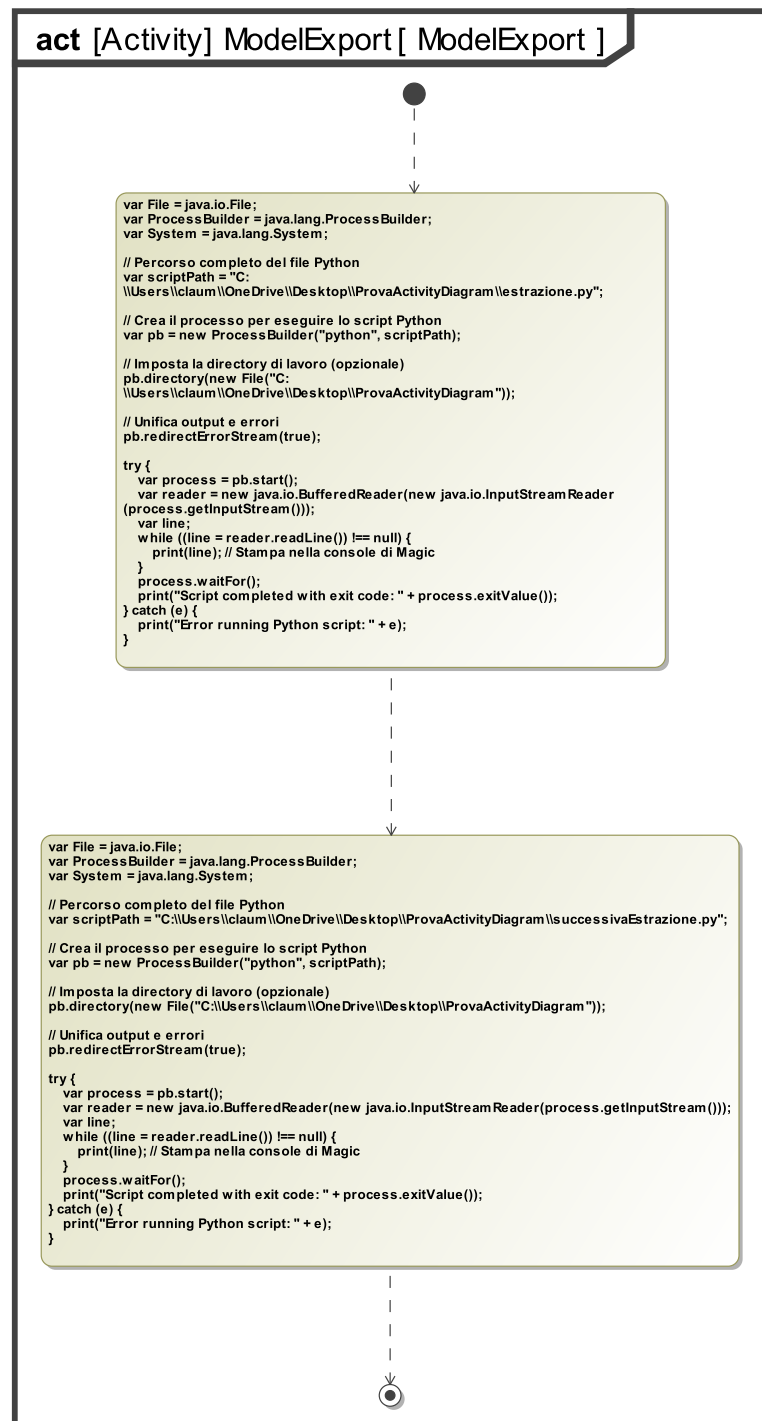


Figure 8. The CATIA MSoSA modeling environment offers a rich and easy-to-use environment. The Activity Diagram shown in this figure creates the JSON file containing the ASTM International substantiation information, ready to be used for all the subsequent calculations.

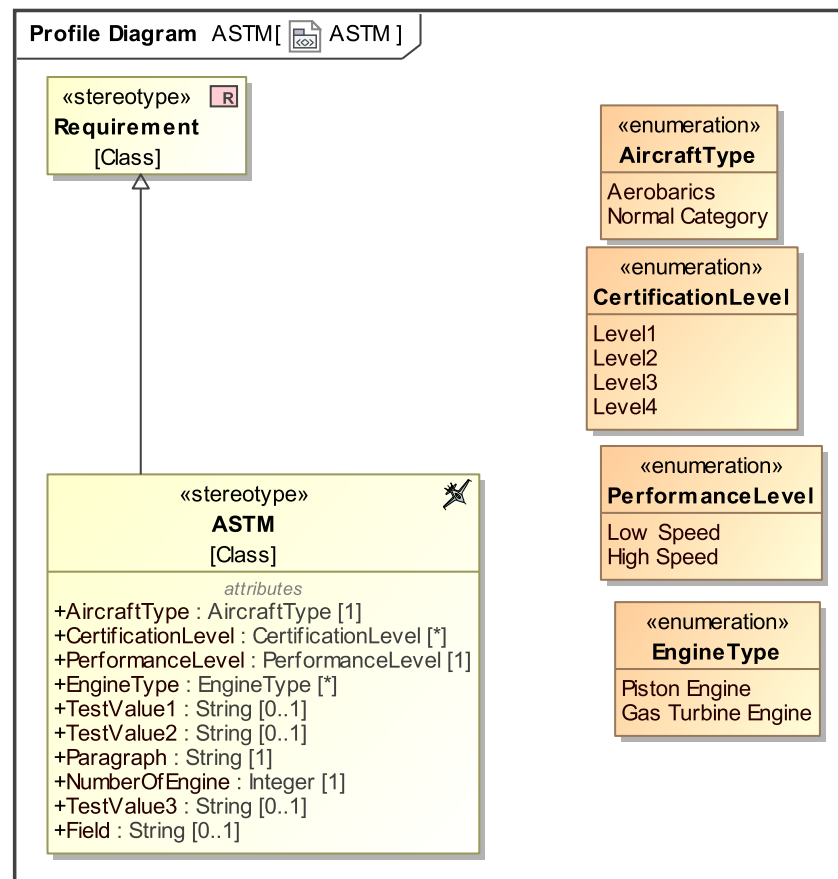


Figure 9. The CATIA MSoSA modeling environment allows the definition of custom SysML Profile Diagrams, such as the one reported in this figure. This stereotype addresses the paragraphs and subparagraphs of the ASTM International documents. The stereotype's properties show the multiplicity symbols indicating the lower and upper bound of the the multiplicity interval, such as [0..1] and [*], where the asterisk indicates an unlimited multiplicity.

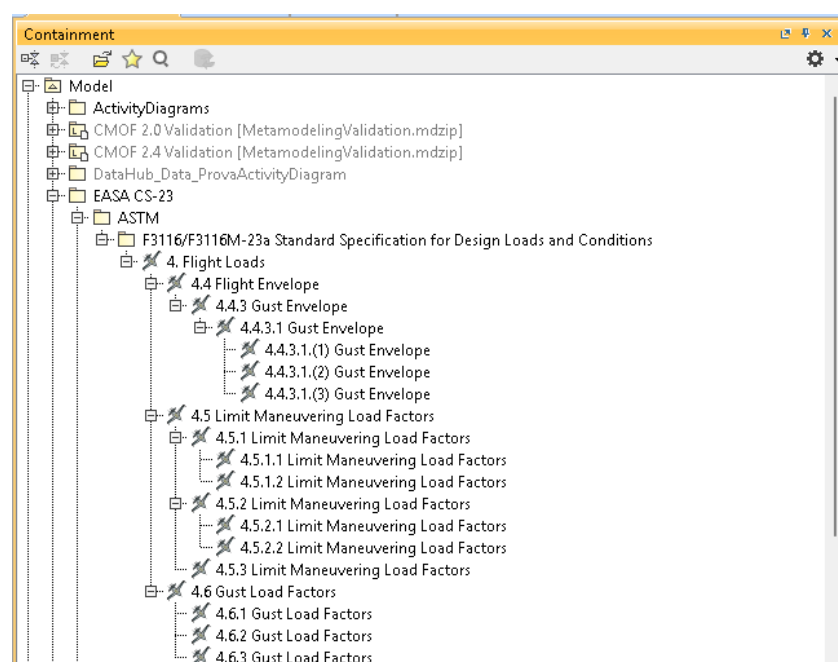


Figure 10. The CATIA MSoSA containment tree allows for organizing the model conveniently, preserving the hierarchical relationships amongst the paragraphs and subparagraphs of the ASTM International documents.

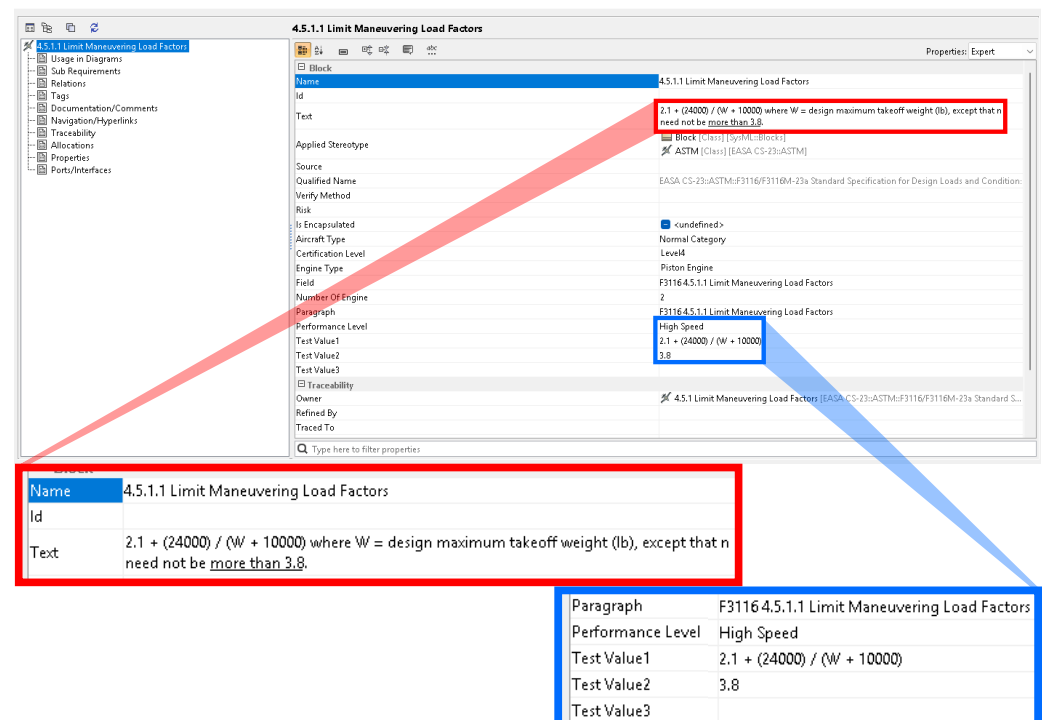


Figure 11. An example of element definition. The paragraph 4.5.1.1 Limit Maneuvering Load Factor from ASTM F3116/3116M Standard Specification for Design Loads and Conditions.

6. Results

This section briefly lists a series of results obtained using the two MBSE approaches for developing a digital certification framework.

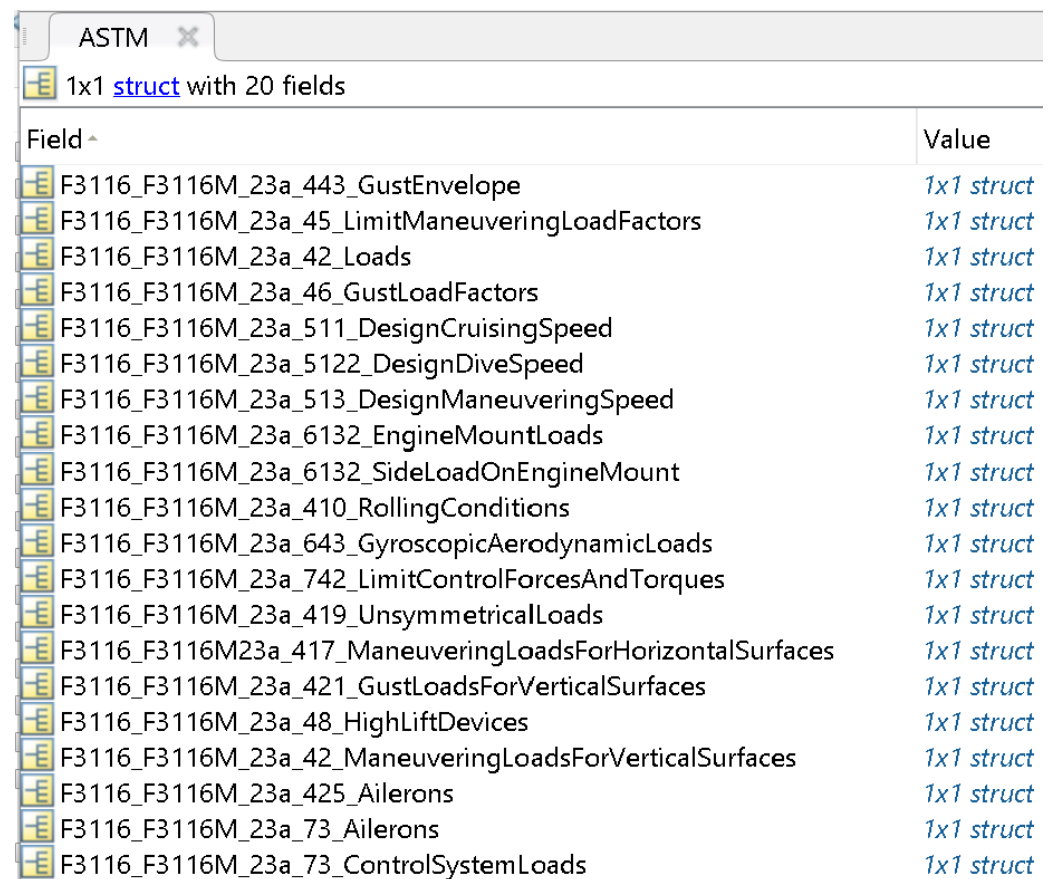
- Section 6.1 illustrates how the model is exported in the MATLAB/Simulink environment as a structured variable. The calculations enable the creation of certification artifacts, such as tables and graphs. The Gust and Maneuvering Flight Envelope diagrams are shown as an example.
- Section 6.2 illustrates the digital certification framework built using CATIA MSoSA. The selected output is, again, the maneuvering and gust flight envelope according to ASTM substantiation requirements and in keeping with the comparison analysis of the present article.

6.1. Eclipse Papyrus and MATLAB/Simulink Results

The MATLAB/Simulink environment has adequate facilities for building structured variables, such as the one shown in Figure 12. The code reads and parses the SysML model and generates the structured variable called ASTM. This structured variable presents the data ordered according to the main section names of the Standard F3116/F3116M.

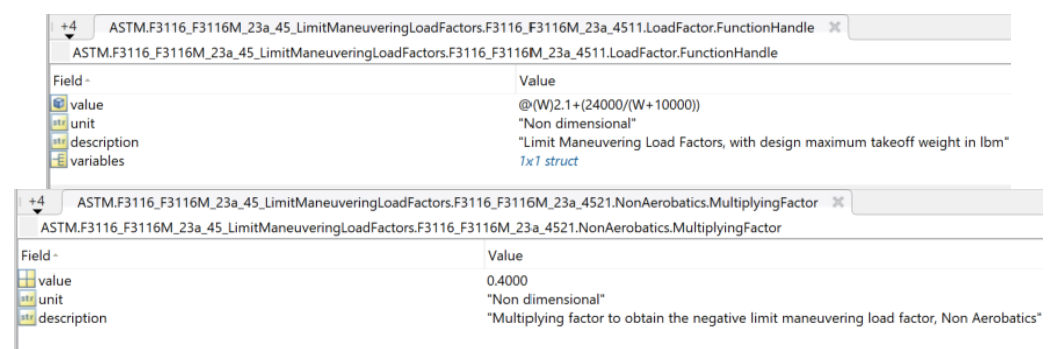
Figure 13 illustrates the content of one of the subfields, specifically the one that concerns the evaluation of the Limit Maneuvering Load Factor. The ASTM variable contains function handles and numerical data ready to be used in all certification-related subsequent calculations.

The MATLAB/Simulink code creates certification artifacts that can be included in a technical certification report. Figure 14 shows the maneuvering and gust flight envelope graphs. These graphs are the most important certification artifacts that are used to show compliance with EASA CS-23 Subpart C high-level requirements.



Field	Value
F3116_F3116M_23a_443_GustEnvelope	1x1 struct
F3116_F3116M_23a_45_LimitManeuveringLoadFactors	1x1 struct
F3116_F3116M_23a_42_Loads	1x1 struct
F3116_F3116M_23a_46_GustLoadFactors	1x1 struct
F3116_F3116M_23a_511_DesignCruisingSpeed	1x1 struct
F3116_F3116M_23a_5122_DesignDiveSpeed	1x1 struct
F3116_F3116M_23a_513_DesignManeuveringSpeed	1x1 struct
F3116_F3116M_23a_6132_EngineMountLoads	1x1 struct
F3116_F3116M_23a_6132_SideLoadOnEngineMount	1x1 struct
F3116_F3116M_23a_410_RollingConditions	1x1 struct
F3116_F3116M_23a_643_GyroscopicAerodynamicLoads	1x1 struct
F3116_F3116M_23a_742_LimitControlForcesAndTorques	1x1 struct
F3116_F3116M_23a_419_UnsymmetricalLoads	1x1 struct
F3116_F3116M23a_417_ManeuveringLoadsForHorizontalSurfaces	1x1 struct
F3116_F3116M_23a_421_GustLoadsForVerticalSurfaces	1x1 struct
F3116_F3116M_23a_48_HighLiftDevices	1x1 struct
F3116_F3116M_23a_42_ManeuveringLoadsForVerticalSurfaces	1x1 struct
F3116_F3116M_23a_425_Ailerons	1x1 struct
F3116_F3116M_23a_73_Ailerons	1x1 struct
F3116_F3116M_23a_73_ControlSystemLoads	1x1 struct

Figure 12. The ASTM structured variable presents the substantiation requirements ordered according to the F3116/F3116M document's main sections.



Field	Value
value	@(W)2.1+(24000/(W+10000))
unit	"Non dimensional"
description	"Limit Maneuvering Load Factors, with design maximum takeoff weight in lbm"
variables	1x1 struct

Field	Value
value	0.4000
unit	"Non dimensional"
description	"Multiplying factor to obtain the negative limit maneuvering load factor, Non Aerobatics"

Figure 13. The ASTM structured variable includes function handles and numerical values, directly extracted from the SysML model, exported as a UML file.

6.2. CATIA MSoSA and C++ Results

Listing 1 shows the JSON file automatically generated using CATIA. It can be seen that the nomenclature used aligns with the user-defined stereotype represented in Figure 4. The file is easily readable by a C++ routine, using the appropriate facilities of modern C++ standards. The JSON content includes numerical parameters and expressions that the C++ algorithms use to assemble and execute the required calculation methods, producing outputs consistent with regulatory expectations. The workflow is therefore internally consistent and follows the ASTM International substantiation instructions accepted by EASA. The C++ code reads the JSON file and produces a structured variable, conceptually similar to the one described in Section 6.1. To start the substantiation calculations for showing compliance with Subpart C, a set of aircraft data is included in a separate JSON file.

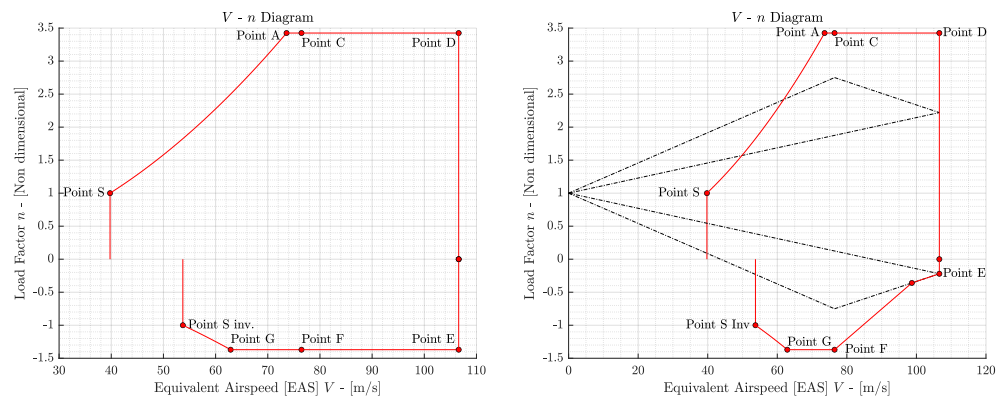


Figure 14. The MATLAB code produces certification artifacts such as tables and graphs. This figure shows the Gust and Maneuvering Flight Envelope as described by the F3116/F3116M Standard.

Listing 1. The ASTM JSON file structures after parsing operations.

```
{
  "F3116 4.5.1.2 Limit Maneuvering Load Factors": {
    "F3116 4.5.1.2 Limit Maneuvering Load Factors": {
      "Paragraph": "F3116 4.5.1.2 Limit Maneuvering Load Factors",
      "TestValue1": 6.0
    }
  },
  "F3116 4.5.1.1 Limit Maneuvering Load Factors": {
    "F3116 4.5.1.1 Limit Maneuvering Load Factors": {
      "Paragraph": "F3116 4.5.1.1 Limit Maneuvering Load Factors",
      "TestValue1": "2.1 + (24000) / (W + 10000)",
      "TestValue2": 3.8
    }
  }
}
```

Figure 15 shows how the outcome of the CATIA MSoSA digital certification framework is a set of certification artifacts, such as the maneuvering and gust flight envelope, which is used as a comparison tool in the present article.

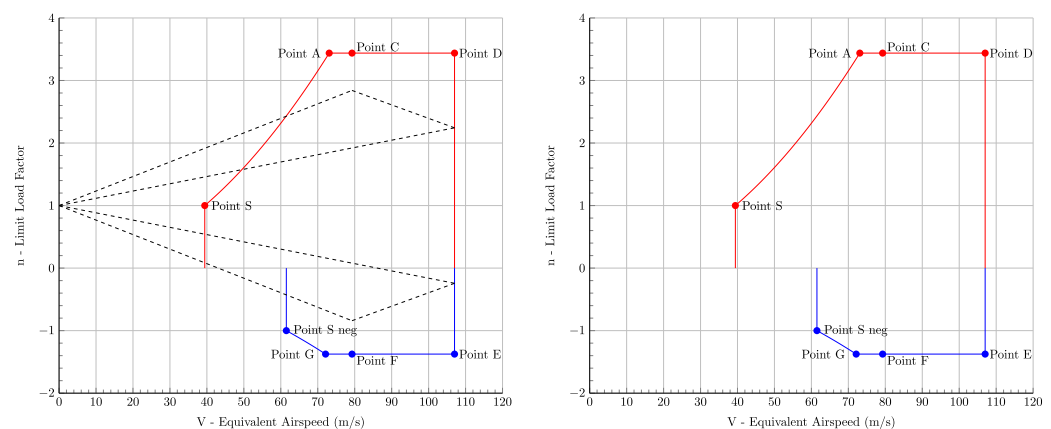


Figure 15. The digital certification framework built with CATIA MSoSA and the C++ code can produce similar certification artifacts, such as the Gust and Maneuvering Flight Envelope shown in this figure.

7. Discussion

Section 6 demonstrates how the two MBSE approaches can be applied to show compliance with high-level EASA CS-23 certification requirements. This section presents a set of observations intended to support the interpretation of the comparative results. Figure 16 provides a qualitative comparison of the two methodologies assessed in this study. Five

parameters are rated on a three-level scale (Low, Medium, High). The five key performance indicators (KPIs) are defined and discussed below.

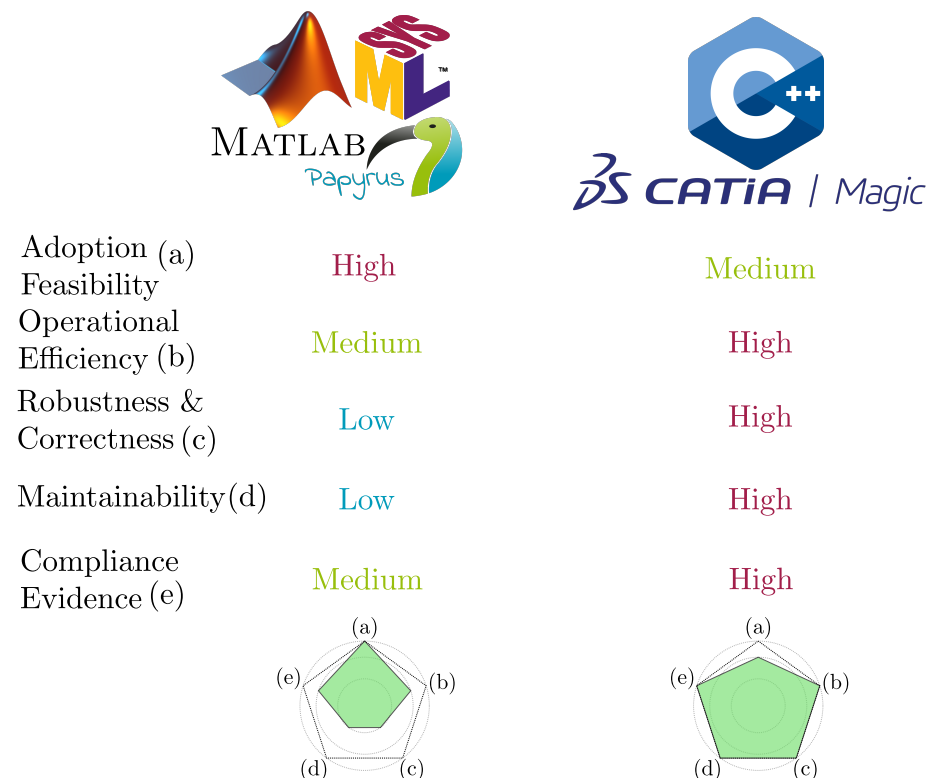


Figure 16. A graphical representation of a subset of *key performance indicators* (KPIs) considered in this study for the two proposed digital certification approaches comparison.

Adoption and Feasibility: This KPI captures the practical feasibility of adopting each methodology and the estimated time required to move from the conceptual design of the digital certification framework to deployment in a practical setting. It qualitatively captures the total development effort needed to implement the framework, including:

- Creating the minimum set of SysML Profile Diagrams and defining the required stereotypes.
- Encoding the certification and substantiation rules in the SysML model.
- Implementing the export, parsing, and use of the SysML regulatory model within the computational workflow. Essentially, this can be reduced on building the companion code suite for working efficiently with the SysML model.

The most time-consuming activity is extracting and formalizing the relevant information from regulatory documents into the SysML model. Once the model is established, the implementation of the associated calculation routines is comparatively straightforward. Eclipse Papyrus is an open-source SysML modeling environment and can be adopted relatively quickly, with adoption costs primarily reflected in the time required to learn the graphical user interface. MATLAB is licensed software and entails acquisition costs; however, it supports rapid development of complex computational workflows for certification tasks. By contrast, CATIA MSoSA involves higher acquisition costs and a steeper learning curve. The C++ implementation can be designed to support robust and reliable substantiation routines; however, these advantages typically require greater development effort. For small aeronautical design and manufacturing organizations, the Eclipse Papyrus/MATLAB framework may be a practical option, particularly when MATLAB is already used within existing

engineering workflows. If needed, Eclipse Papyrus can be extended using model-transformation languages such as Epsilon [40]. However, the adoption of custom automation plugins should be justified through a careful, upfront assessment of the intended application. Adopting CATIA MSoSA is technically feasible but may represent a substantial investment for such organizations. Although C++ is widely used in safety-critical software development, integrating and maintaining a C++ backend still requires specialized expertise and process maturity. Overall, the organizational changes required to adopt an MBSE-based certification toolchain are feasible, but they typically demand adjustments to established management and execution practices, which may be less readily implemented in smaller organizations.

Operational Efficiency: This KPI measures the degree of automation that can be achieved in practice using the digital certification framework. The rationale for a digital certification framework is to reduce, and where possible eliminate, human intervention across certification steps, because many coherence and traceability gaps, inconsistencies, and errors arise during manual information exchanges and manual edits. Eclipse Papyrus does not provide native facilities for executing certification workflows within the modeling environment; however, a satisfactory level of automation can be achieved through external tools, such as MATLAB. By contrast, CATIA MSoSA supports integrated workflow execution through Activity Diagrams, enabling many certification steps to be performed automatically. In practice, the user can trigger predefined procedures that execute certification and substantiation tasks without direct interaction with intermediate data exchanges. Once the certification artifacts are generated, the environment provides an efficient report-generation capability that can be configured through templates aligned with the certification task under consideration. Ideally, the framework would automate the entire procedure in a practical setting; however, during early adoption, the effort required to create, update, and maintain the digital certification database may exceed the framework's operational capacity. For this reason, Operational Efficiency is included as a KPI in this study.

Robustness and Correctness: This KPI assesses the reliability of the digital certification framework under changes to EASA regulations and ASTM substantiation documents. It also reflects the extent to which regulatory content is correctly interpreted and translated into consistent certification artifacts. The Eclipse Papyrus/MATLAB implementation relies on a comparatively fragile—under substantial changes of the reference documents—parsing approach based on regular expressions applied to the exported SysML/UML representation, which contains information transcribed from the regulatory sources. Substantial changes in regulatory text or structure can therefore reduce stability. Although the MATLAB parser was developed with error handling, fail-safe behavior and safeguards against silent failures, the workflow remains sensitive to variations in the underlying rule content. As discussed below, this sensitivity also affects maintainability. By contrast, the CATIA MSoSA implementation extracts regulatory content through user-defined SysML stereotypes. The Python companion routine first identifies model elements that carry the relevant stereotypes and then exports only non-empty fields to the JSON input, reducing dependence on text–pattern matching. This approach improves robustness and increases the likelihood of successful execution; it also supports usability and maintainability. In practice, failures are expected primarily when the regulatory structure changes in ways that invalidate the modeled hierarchy (e.g., major restructuring of paragraphs and subparagraphs or changes that require a different modeling schema).

Maintainability: This KPI evaluates how easily the digital certification framework can be maintained under operational conditions. As discussed above, the Eclipse Papyrus/MATLAB toolchain is comparatively less stable and generally requires greater maintenance effort to remain fully operational. Its structure is less straightforward, and individual maintenance tasks often demand more effort. For example, when certification regulations change significantly, the parsing routines must be verified against the updated regulatory text and subsequently revalidated, which can be time-consuming and cumbersome. By contrast, the CATIA MSoSA/C++ implementation is more robust and typically requires fewer maintenance activities. Its parsing mechanism is inherently more stable because it relies on structured SysML stereotypes rather than direct manipulation of regulatory text. Once the SysML content has been updated, the framework can accommodate substantial changes in the requirements without extensive modifications to the parsing logic. In addition, the framework components are modular and self-contained, which supports independent updates and contributes to improved maintainability under normal operating conditions.

Compliance Evidence: This KPI assesses the extent to which the digital certification framework supports the compliance process by producing complete and usable certification artifacts. In the MATLAB-based implementation, traceability between requirements, calculations, and generated outputs is comparatively more limited, although the workflow can generate certification reports that include figures, tables, and regulatory excerpts, as shown in ref. [13]. By contrast, the CATIA MSoSA environment provides a more integrated view of the certification status and the associated artifacts. Its report wizard can be configured to incorporate generated data consistently, supporting coherent assembly of figures, tables, and narrative elements. As a result, artifact generation is automated and their inclusion in certification reports is more straightforward. Moreover, when the supporting C++ routines are designed to preserve links between artifacts and the corresponding certification requirements, these links can be used to drive report generation and further strengthen traceability.

The present work is limited in scope, and the evaluation of the CATIA MSoSA report wizard was not sufficiently thorough to be included. The primary objective was to establish and demonstrate a digital certification capability within the CATIA MSoSA environment, which this study achieves. Future work will present a broader digital certification framework implemented in CATIA MSoSA and supported by a Python-based substantiation library for generating certification artifacts. An Eclipse Papyrus/MATLAB digital certification framework has already been presented in ref. [13]. The scope of this study is limited to EASA CS-23, Subpart C; however, this limitation does not preclude the scalability of the proposed methodologies to other subparts. In particular, enhanced traceability mechanisms and the integration of verification and validation capabilities represent transferable advantages that can be applied across the broader aircraft certification process. The baseline for comparison remains the classic document-based approach, as discussed in Section 4. Aeronautical regulators currently assess compliance primarily through certification artifacts such as tables, figures, and technical reports; therefore, these outputs cannot be removed from the certification workflow at present. Instead, the proposed digital certification frameworks can be used to improve traceability and robustness and to reduce manual effort in producing and maintaining certification evidence. Several final considerations are in order:

- Regardless of whether the MBSE digital certification framework is implemented using Eclipse Papyrus with MATLAB/Simulink or CATIA MSoSA with C++ executables, the advantages of MBSE techniques in general aviation aircraft certification are evident. The automated process enables requirement and artifact traceability, significantly reducing errors and minimizing the risk of mistakes. These benefits arise because the

calculation methods and the overall procedure adhere to the EASA and FAA acceptable means of compliance, thereby enabling a digital certification database that is accessible, maintainable, and adaptable to the aircraft architecture. As a result, the certification process can be improved in coherence, consistency, and clarity, allowing applicants to exchange information with regulators more effectively. However, as in many complex industrial contexts, acquiring the required technologies and implementing an MBSE-based certification toolchain can be costly and may be unaffordable for small design organizations. Moreover, the technical complexity of the toolchain can create a knowledge barrier within design and manufacturing departments.

- Both MATLAB/Simulink and CATIA MSoSA involve significant licensing costs, which may hinder adoption by small- or medium-sized aeronautical design and manufacturing companies. For industry-level applications, an optimal solution could be the use of CATIA MSoSA integrated with a suite of C++ tools within its digital certification workflow. In this configuration, C++ executables operate in the background, while the user interacts exclusively at the system design level through the SysML model. This separation allows applicants to train personnel on the SysML modeling environment alone while treating the companion codebase as a backend component that can be managed and maintained internally or outsourced as a technical service. CATIA MSoSA also provides a convenient report wizard for creating certification artifacts, including technical reports with graphs, diagrams, formulas, tables, and other relevant data. For smaller organizations typical of the general aviation sector, adopting MATLAB/Simulink to implement MBSE frameworks in combination with other digital tools may be less demanding, particularly where MATLAB expertise is already available. However, such toolchains may offer fewer task-specific capabilities than an integrated environment such as CATIA MSoSA.
- The MATLAB/Simulink implementation is less stable because the SysML model, created with a less capable tool such as Eclipse Papyrus, offers fewer features and is therefore less easily explored. Regular expressions effectively link requirements to substantiation calculations; however, significant amendments to ASTM documents may cause the code to fail when parsing the UML file. In the best case, the code will generate an error; in the worst case, it may run silently without signaling any issues during parsing. The CATIA MSoSA implementation avoids this problem by using richer stereotypes, as discussed in Section 5.2. Instead of relying on regular expressions, the Python companion code scans the UML file for model elements with non-empty TestValue fields, creating a JSON input file for the C++ executables. This approach is inherently more stable and easier to maintain because it is embedded within the CATIA MSoSA environment.

This discussion suggests that the MBSE digital certification framework implemented in CATIA MSoSA and supported by a C++ code suite is both feasible and responsive to the practical needs of applicants, certification engineers, and certification specialists. The toolchain can automate substantial portions of the certification workflow, and, although this study addresses only EASA CS-23, Subpart C, the underlying approach is potentially scalable to other subparts [13], including those concerned with systems, engines, propellers, and related domains. CATIA MSoSA provides a flexible, ready-to-use environment for integrated workflow execution, whereas Eclipse Papyrus offers a viable open-source alternative for SysML modeling but provides more limited support for orchestrating and executing end-to-end certification workflows within the modeling environment.

8. Conclusions

This study examines the application of MBSE toolchains to address challenges in aircraft certification. Airworthiness requirements were digitalized using SysML, which enables structured encoding of certification data and associated metadata. The primary purpose of this study is to present two MBSE-based methodologies for developing a digital certification framework, thereby providing a reference for examining and comparing alternative approaches to this complex engineering process. A qualitative comparison of the two methodologies has been presented.

The first approach employs Eclipse Papyrus as the modeling environment, integrated with MATLAB/Simulink for parsing and computation. This configuration allows rapid certification-related calculations by generating structured variables populated with data derived from formal specifications.

The second approach leverages CATIA Magic Systems of Systems Architect (MSoSA), which automates the certification workflow through Activity Diagrams and orchestrates calculations using C++ executables. Both implementations support the generation of certification artifacts and comprehensive reports, either through MATLAB's API or document templates within CATIA MSoSA.

Future work will focus on validating these frameworks in industrial settings using quantitative Key Performance Indicators (KPIs) and improving robustness by refining the relationship between modeling and parsing phases.

Author Contributions: Conceptualization, C.M. and M.T.; methodology, C.M.; software, C.M.; validation, C.M. and P.D.V.; formal analysis, C.M.; investigation, C.M.; resources, P.D.V.; data curation, C.M.; writing—original draft preparation, C.M.; writing—review and editing, C.M. and P.D.V.; visualization, C.M.; supervision, P.D.V.; project administration, P.D.V.; funding acquisition, P.D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This article is part of the DIGACE research project that has received financial support from the Italian Ministry of University and Research through the PRIN 2022 funding initiative (Grant No. 20229H8N9P).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors thank Giuseppe Di Gironimo and Francesco Marino for their guidance and support in the use of modeling tools, including CATIA NoMagic Systems of Systems Architect.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CS-23	Certification Specifications for Part 23 aircraft
CS-VLA	Certification Specifications for Very Light Aircraft
AMC	Acceptable Means of Compliance
MBSE	Model-Based Systems Engineering
SysML	Systems Modeling Language
UML	Unified Modeling Language
ASTM	American Society for Testing and Materials
DIGACE	Digital Innovative General Aviation Aircraft Certification

POLIMI	Politecnico di Milano
POLITO	Politecnico di Torino
UNINA	University of Naples
MoC	Means of Compliance
TC	Type Certificate
MTOW	Maximum Takeoff Weight
INCOSE	International Council on Systems Engineering
API	Application Programming Interface
CCC	Certification Compliance Checklist
CDD	Certification Demonstration Data
CDI	Certification Demonstration Items
CDA	Certification Demonstration Artifacts
JSON	JavaScript Object Notation
XML	Extensible Markup Language
CS	Certification Specifications
FAR	Federal Aviation Regulations
FAA	Federal Aviation Administration
EASA	European Union Aviation Safety Agency
MSoSA	Magic Systems of Systems Architect
PRIN	Progetti di Rilevante Interesse Nazionale
KPI	Key Performance Indicator
SI	Système international

References

1. Xie, J.; Harrison, E.D.; Mavris, D.N. Quantifying Impacts of Uncertainties on Certification-Driven Design. *J. Aircr.* **2024**, *61*, 1354–1377. [CrossRef]
2. Schlickemaier, H.; Voss, M.G.; Wilkinson, R.E. Certification Rules and Standards Review. NASA Technical Report 2019. Available online: <https://ntrs.nasa.gov/citations/20190033235> (accessed on 15 March 2026).
3. Bleu-Laine, M.H.; Bendarkar, M.V.; Xie, J.; Briceno, S.I.; Mavris, D.N. A model-based system engineering approach to normal category airplane airworthiness certification. In *Proceedings of the AIAA Aviation 2019 Forum*; AIAA: Reston, VA, USA, 2019; p. 3344.
4. Bendarkar, M.V.; Xie, J.; Briceno, S.I.; Harrison, E.D.; Mavris, D. A model-based aircraft certification framework for normal category airplanes. In *Proceedings of the AIAA Aviation 2020 Forum*; AIAA: Reston, VA, USA, 2020; p. 3096.
5. Mauery, T.; Alonso, J.; Cary, A.; Lee, V.; Malecki, R.; Mavriplis, D.; Medic, G.; Schaefer, J.; Slotnick, J. *A Guide for Aircraft Certification by Analysis*; Technical report; National Aeronautics and Space Administration: Washington, DA, USA, 2021.
6. Jezegou, J.; Blondel de Joigny, C.; Bureau, V.; Seguin, C.; Jiménez Carrasco, B.; André, R.; Cilio, G.; Simone, E. Certification Readiness Level Scale: Maturing the Certifiability of Innovative Aircraft. In *Proceedings of the AIAA SCITECH 2026 Forum*; AIAA: Reston, VA, USA, 2026; p. 1385.
7. Xie, J.; Cai, Y.; Sarojini, D.; Harrison, E.; Mavris, D.N. Vertical Tail Sizing and Power Split Optimization for the PEGASUS Concept Considering Certification Requirements. In *Proceedings of the AIAA Aviation 2022 Forum*; AIAA: Reston, VA, USA, 2022; p. 3204.
8. Sarojini, D.; Xie, J.; Cai, Y.; Corman, J.A.; Mavris, D. A Certification-Driven Platform for Multidisciplinary Design Space Exploration in Airframe Early Preliminary Design. In *AIAA Aviation 2020 Forum*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2020; Chapter AIAA Aviation 2020 Forum; p. 3157. [CrossRef]
9. European Union Aviation Safety Agency (EASA). Day 1: Part 21 Workshop at EASA Headquarters; Day 2: Certification Conference at the Cologne Botanical Garden, with On-site and Live-Streamed Participation. In *Proceedings of the EASA Part 21 Workshop and Certification Conference 2026*, Hybrid, 10–11 March 2026. Available online: <https://www.easa.europa.eu/en/newsroom-and-events/events/easa-part-21-workshop-and-certification-conference-2026> (accessed on 15 March 2026).
10. Xie, J.; Bendarkar, M.V.; Cai, Y.; Mavris, D.N. Evaluating “Critical-Engine-Inoperative” Conditions for Distributed Hybrid-Electric Aircraft. *J. Aircr.* **2025**, *63*, 1–18. [CrossRef]
11. Xie, J.; Bendarkar, M.V.; Cai, Y.; Mavris, D.N. Revisiting “Critical-Engine-Inoperative” Conditions for Distributed Hybrid-Electric Aircraft. In *Proceedings of the AIAA Aviation 2023 Forum*; AIAA: Reston, VA, USA, 2023; p. 3608.

12. Cardoso, S.H.S.B.; Oliveira, M.V.R.d.; Godoy, J.R.S. eVTOL certification in FAA and EASA performance-based regulation environments: A bird strike study-case. *J. Aerosp. Technol. Manag.* **2022**, *14*, e2122. [\[CrossRef\]](#)
13. Vecchia, P.D.; Mirabella, C.; Tuccillo, M.; Riboldi, C.E.; Fioriti, M.; Shahini, F.; Roncolini, F. A SysML-based framework towards EASA CS-23 digitalization: An MBSE approach. *Transp. Res. Interdiscip. Perspect.* **2026**, *36*, 101810. [\[CrossRef\]](#)
14. Mirabella, C.; Tuccillo, M.; Vecchia, P.D. A model-based systems engineering digital certification framework for general aviation aircraft. *Aerotec. Missili Spaz.* **2024**, *104*, 371–386. [\[CrossRef\]](#)
15. Bottero, M.; Gualeni, P. Systems Engineering for Naval Ship Design Evolution. *J. Mar. Sci. Eng.* **2024**, *12*, 210. [\[CrossRef\]](#)
16. Gaignebet, A.; Chapurlat, V.; Zacharewicz, G.; Richet, V.; Plana, R. A Model Based System Commissioning Approach for Nuclear Facilities. *Sustainability* **2021**, *13*, 10520. [\[CrossRef\]](#)
17. Xie, J.; Chakraborty, I.; Briceno, S.I.; Mavris, D.N. Development of a certification module for early aircraft design. In *Proceedings of the AIAA Aviation 2019 Forum*; AIAA: Reston, VA, USA, 2019; p. 3576.
18. Campo, K.X.; Teper, T.; Eaton, C.E.; Shipman, A.M.; Bhatia, G.; Mesmer, B. Model-based systems engineering: Evaluating perceived value, metrics, and evidence through literature. *Syst. Eng.* **2023**, *26*, 104–129. [\[CrossRef\]](#)
19. Singh, V.; Willcox, K.E. Engineering Design with Digital Thread. *AIAA J.* **2018**, *56*, 4515–4528. [\[CrossRef\]](#)
20. Fakhraian, E.; Semanjski, I.; Semanjski, S.; Aghezzaf, E.H. Towards safe and efficient unmanned aircraft system operations: Literature review of digital twins' applications and european union regulatory compliance. *Drones* **2023**, *7*, 478. [\[CrossRef\]](#)
21. Kausch, H.; Pfeiffer, M.; Raco, D.; Rumpe, B.; Schweiger, A. Model-driven development for functional correctness of avionics systems: A verification framework for SysML specifications. *CEAS Aeronaut. J.* **2025**, *16*, 33–48. [\[CrossRef\]](#)
22. De Florio, F. *Airworthiness: An Introduction to Aircraft Certification and Operations*; Butterworth-Heinemann: Oxford, UK, 2016.
23. Cook, S.P. Digital airworthiness certification: Opportunities for unmanned aircraft systems. In *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1432–1437.
24. Cartile, A.; Marsden, C.; Liscouët-Hanke, S. Digital Transformation in Aircraft Design and Certification: Developing Requirements for Modeling Regulatory Documentation. *Aerospace* **2025**, *12*, 724. [\[CrossRef\]](#)
25. Wilking, F.; Horber, D.; Goetz, S.; Wartzack, S. Utilization of system models in model-based systems engineering: Definition, classes and research directions based on a systematic literature review. *Des. Sci.* **2024**, *10*, e6. [\[CrossRef\]](#)
26. Madni, A.M.; Purohit, S. Economic analysis of model-based systems engineering. *Systems* **2019**, *7*, 12. [\[CrossRef\]](#)
27. Li, L.; Soskin, N.L.; Jbara, A.; Karpel, M.; Dori, D. Model-based systems engineering for aircraft design with dynamic landing constraints using object-process methodology. *IEEE Access* **2019**, *7*, 61494–61511. [\[CrossRef\]](#)
28. Wang, Y.; Zhang, A.; Bi, W.; Han, Y.; Li, H. Design method of civil aircraft based on MBSE with improved process model. *Chin. J. Aeronaut.* **2025**, *38*, 103696. [\[CrossRef\]](#)
29. Liscouët-Hanke, S. A Model-Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures. Ph.D. Thesis, Institut National des Sciences Appliquées de Toulouse, Toulouse, France, 2008.
30. Liscouët-Hanke, S.; Jahanara, H.; Bauduin, J.L. A Model-Based Systems Engineering Approach for the Efficient Specification of Test Rig Architectures for Flight Control Computers. *IEEE Syst. J.* **2020**, *14*, 5441–5450. [\[CrossRef\]](#)
31. Torrigiani, F.; Deinert, S.; Fioriti, M.; Fede, F.D.; Jungo, A.; Pisu, L.; Sanchez, F.; Liscouët-Hanke, S.; Ciampa, P.D.; Nagel, B. An MBSE Certification-Driven Design of a UAV MALE Configuration in the AGILE 4.0 Design Environment. In *AIAA Aviation 2021 Forum*; AIAA: Reston, VA, USA, 2021. [\[CrossRef\]](#)
32. Jeyaraj, A.K.; Tabesh, N.; Liscouët-Hanke, S. Connecting Model-based Systems Engineering and Multidisciplinary Design Analysis and Optimization for Aircraft Systems Architecting. In *AIAA Aviation 2021 Forum*; AIAA: Reston, VA, USA, 2021. [\[CrossRef\]](#)
33. Meijer, J.J. Aeroelasticity. In *Introduction to Flight Test Engineering*; RTO AGARDograph 300 Flight Test Techniques Series; Stoliker, F.N., Ed.; NATO Research and Technology Organisation (RTO): Neuilly-sur-Seine, France, 2005; Volume 14, pp. 14-1–14-13.
34. Nabi, H.; Lombaerts, T.; Zhang, Y.; Van Kampen, E.; Chu, Q.; de Visser, C.C. Effects of structural failure on the safe flight envelope of aircraft. *J. Guid. Control. Dyn.* **2018**, *41*, 1257–1275. [\[CrossRef\]](#)
35. *ASTM F3116/F3116M-24*; Standard Specification for Design Loads and Conditions. ASTM International Standard: West Conshohocken, PA, USA, 2025. [\[CrossRef\]](#)
36. *ASTM F3396/F3396M-23a*; Standard Practice for Aircraft Simplified Loads Criteria. ASTM International Standard: West Conshohocken, PA, USA, 2023. [\[CrossRef\]](#)
37. *ASTM F3331-18(R2023)*; Standard Practice for Aircraft Water Loads. ASTM International Standard: West Conshohocken, PA, USA, 2023. [\[CrossRef\]](#)
38. *ASTM F2245-23*; Standard Specification for Design and Performance of a Light Sport Airplane. ASTM International Standard: West Conshohocken, PA, USA, 2023. [\[CrossRef\]](#)

39. Papke, B.; Hause, M.; Hetherington, D.; McGervey, S.; Rodriguez, S. MBSE Model Management Pain Points—Wait, this looks familiar! In Proceedings of the INCOSE International Symposium, Honolulu, HI, USA, 15–20 July 2023. [[CrossRef](#)]
40. Kolovos, D.S.; Paige, R.F.; Polack, F.A. The Epsilon Transformation Language. In *Proceedings of the 1st International Conference on Theory and Practice of Model Transformations*; ICMT '08; Springer: Berlin/Heidelberg, Germany, 2008; pp. 46–60. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.