Article

Not peer-reviewed version

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

Claudio Mirabella *, Michele Tuccillo, Pierluigi Della Vecchia

Posted Date: 18 March 2026

doi: 10.20944/preprints202603.1390.v1

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

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, Naples, Italy
* Correspondence: claudio.mirabella@unina.it; Tel.: +39-338-471-3099

Abstract

This study addresses the growing complexity of aircraft certification by introducing Model-Based Systems Engineering (MBSE) techniques within the Digital Innovative General Aviation Aircraft Certification (DIGACE) project. The research aims to digitalize airworthiness requirements and means of compliance, focusing on EASA CS-23, Amendment 6. The proposed framework employs SysML for formal modeling and MATLAB/Simulink for automated processing, enabling structured encoding of certification data and metadata. The SysML-based UML model highlights inconsistencies in current regulatory standards while supporting automated generation of certification artifacts and technical reports. Two implementations are examined: one using Eclipse Papyrus in conjunction with MATLAB for parsing and computation, and another leveraging CATIA Magic Systems of Systems Architect to execute certification-related tasks through Activity Diagrams and C++ routines. Both approaches demonstrate the feasibility of automating certification workflows, reducing manual effort, and improving traceability. These findings suggest that MBSE methodologies can significantly enhance certification efficiency, support the integration of innovative technologies, and maintain compliance with legacy safety standards. Future work will focus on validating these frameworks in industrial settings, and refining their robustness for broader adoption.

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 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.

 doi:10.20944/preprints202603.1390.v1

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 [9,10]. This may lead to a decrease in competitiveness and innovation over time in the aeronautical industry, especially in the General Aviation market [11].

To address these issues, 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 [12,13].

SysML provides a robust formalism for managing this complexity. The current digital certification framework, developed using SysML and MATLAB/Simulink, demonstrates its effectiveness through two implementations: one based on Eclipse Papyrus and another using CATIA Magic Systems of Systems Architect (MSoSA). These approaches enable automated parsing, calculation, and generation of certification artifacts, offering a streamlined process for manufacturers and regulatory agencies. The adoption of MBSE methodologies promises to enhance certification efficiency [14], support innovative technologies [15], and maintain safety margins even for unconventional designs [16,17], at least at the certification level.

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 [18].

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 [19].

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.

2. 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 2.1 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 2.2 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.

2.1. DIGACE Implementation

The DIGACE implementation consists of three phases designed to digitalize certification specifications compliance verification.

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

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 March 2026 doi:10.20944/preprints202603.1390.v1

textual information extracted from EASA CS-23 and Means of Compliance (MoC) documents, along with metadata describing relationships, stereotypes, and model elements.

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*.

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.

This methodology enables automated extraction, calculation, and reporting, reducing manual effort and improving accuracy while maintaining compliance with regulatory standards.

2.2. CATIA MSoSA Implementation

The CATIA Magic Systems of Systems Architect (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.

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*—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 fiels such as TestExpressions and TestValues:

- TestExpressions is 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 model's elements.
- TestValues is 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.

It is easy to instruct the Python routine to search for model's elements having non-empty TestExpressions and TestValues 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 2.1 are circumvented efficiently, with an increase in execution and results stability.

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.

3. Results

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

- Section 3.1 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 3.2 illustrates the digital certification framework built using CATIA MSoSA. The selected output is, again, the Gust and Maneuvering Flight Envelope according to ASTM substantiation requirements and in keeping with the comparison analysis of the present article.

3.1. Eclipse Papyrus and MATLAB/Simulink Results

The MATLAB/Simulink environment has adequate facilities for building structured variables, such as the one shown in Figure 1. 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/F3115M.

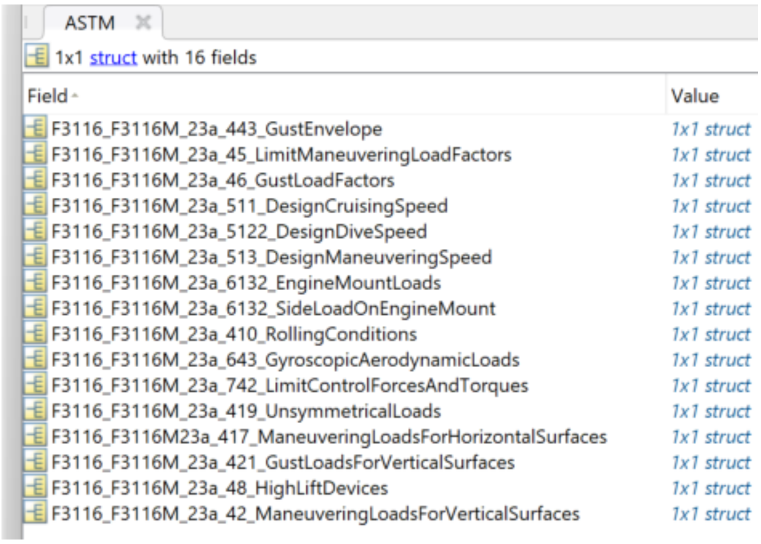


Figure 1. The ASTM structured variable presents the substantiation requirements ordered according to the F3116/F3116M document's main sections.

Figure 2 illustrates the content of one of the subfields, and 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.

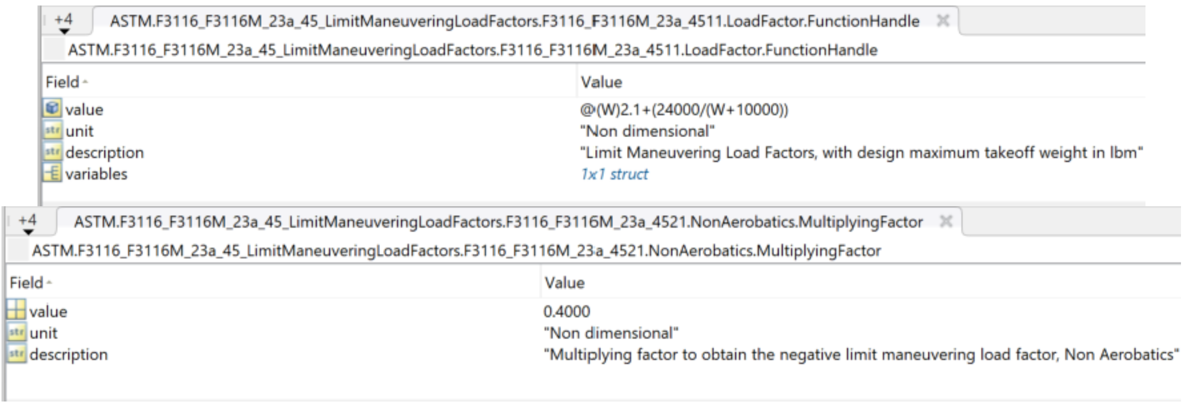


Figure 2. The ASTM structured variable includes function handles and numerical values, directly extracted from the SysML model, exported as a UML file.

The MATLAB/Simulink code creates certification artifacts that can be included in a technical certification report. Figure 3 shows the Flight Envelope and the Gust and Maneuvering 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.

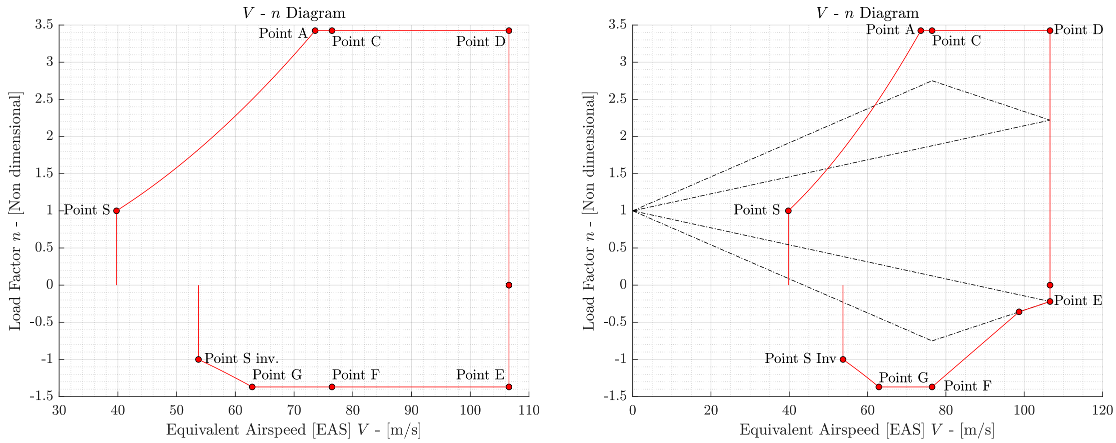


Figure 3. The MATLAB code produces certification artifacts such as tables and graphs. This figure shows the Gust and Maneuvering Flight Envelope as described by the F3116/F3116M Standard.

3.2. CATIA MSoSA and C++ Results

Figure 4 shows the CATIA MSoSA environment. The Activity Diagram used to create the JSON input file and launch the C++ executables is visible on the right. On the left, the containment tree illustrates the hierarchical structure of the model, a tree structure necessary for capturing the paragraphs and subparagraphs subdivision of the ASTM documents.

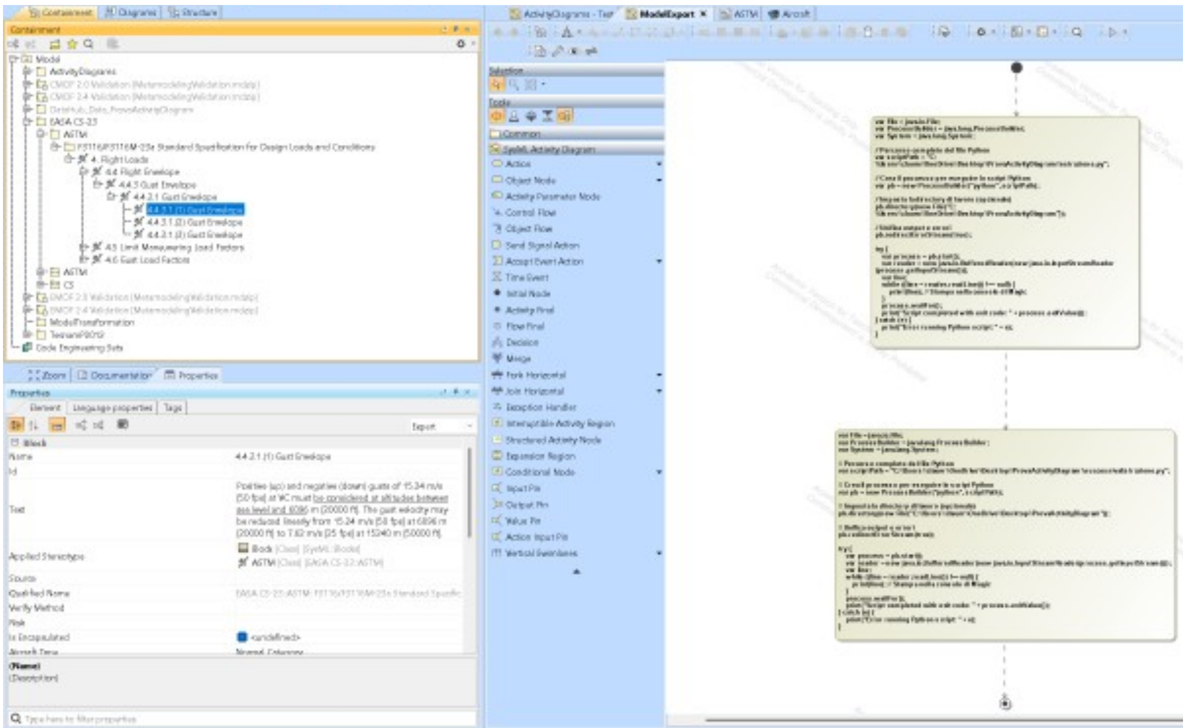
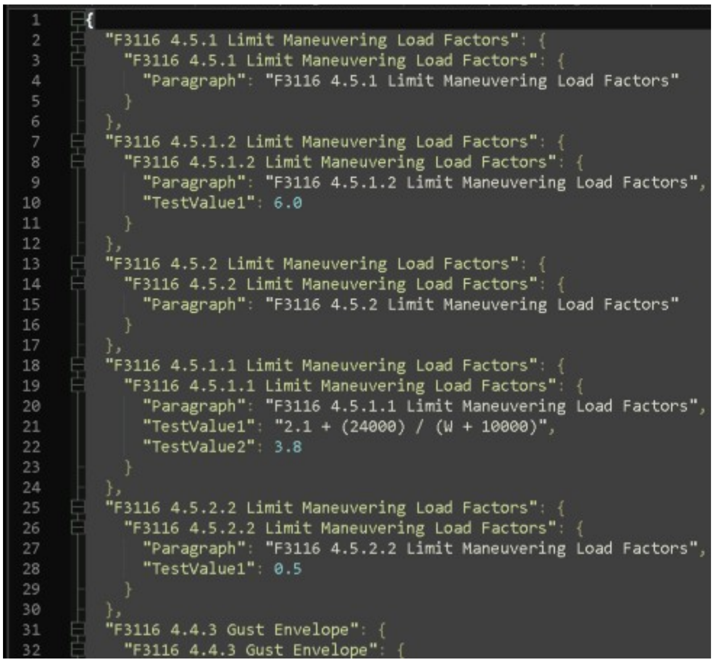


Figure 4. The CATIA MSoSA environment, with the Activity Diagram used to prepare the JSON input file and launch external C++ routines.

Figure 5 shows the JSON file automatically generated using CATIA. The file is easily readable by a C++ routine, using the appropriate facilities of modern C++ standards. The C++ code reads the JSON file and produces a structured variable, conceptually similar to the one described in Section 3.1. To start the substantiation calculations for showing compliance with Subpart C, a set of aircraft data is included in a separate JSON file.



```
{
  "F3116 4.5.1 Limit Maneuvering Load Factors": {
    "F3116 4.5.1 Limit Maneuvering Load Factors": {
      "Paragraph": "F3116 4.5.1 Limit Maneuvering Load Factors"
    }
  },
  "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 4.5.2 Limit Maneuvering Load Factors": {
    "F3116 4.5.2 Limit Maneuvering Load Factors": {
      "Paragraph": "F3116 4.5.2 Limit Maneuvering Load Factors"
    }
  },
  "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
    }
  },
  "F3116 4.5.2.2 Limit Maneuvering Load Factors": {
    "F3116 4.5.2.2 Limit Maneuvering Load Factors": {
      "Paragraph": "F3116 4.5.2.2 Limit Maneuvering Load Factors",
      "TestValue1": 0.5
    }
  },
  "F3116 4.4.3 Gust Envelope": {
    "F3116 4.4.3 Gust Envelope": {
```

Figure 5. The input JSON file that collects all the substantiation information in a convenient, structured fashion.

Figure 6 shows how the outcome—in terms of certification artifacts—of the CATIA MSoSA digital certification framework is a set of certification artifacts, such as the Gust and Maneuvering Flight Envelope used as a comparison tool in the present article.

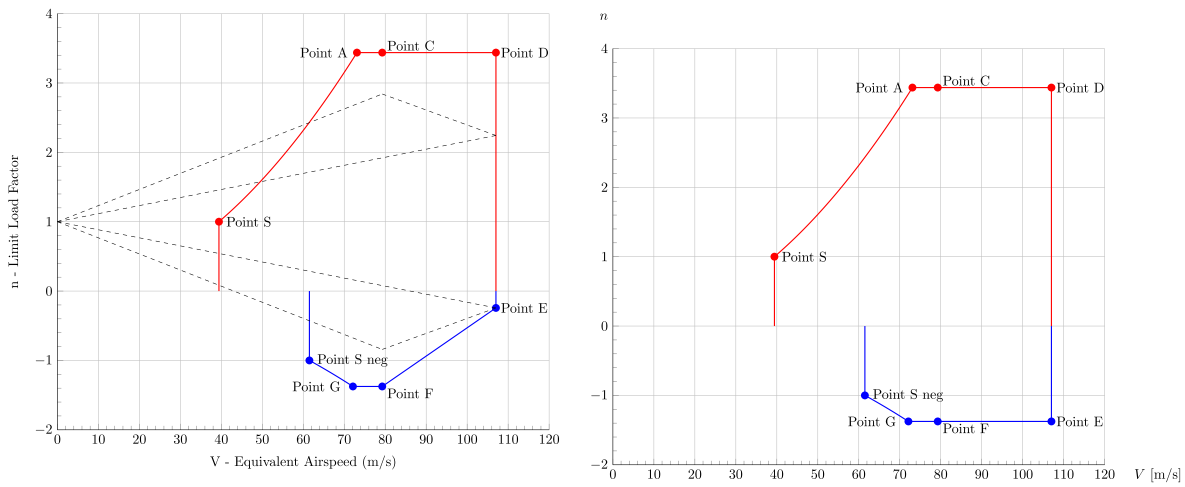


Figure 6. The digital certification framework built with CATIA MSoSA and the C++ code can produce similar certification artifacts, such as the Gust and Maneuvering Flight Envelope shown in this figure.

4. Discussion

Section 3 demonstrated how the two MBSE approaches can be applied to show compliance with high-level EASA CS-23 certification requirements. This section presents a series of observations intended to help readers interpret and understand the results of the comparison conducted in this study.

- 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

 doi:10.20944/preprints202603.1390.v1

in general aviation aircraft certification are evident. The automated process enables requirement and artifact traceability, significantly reducing errors and minimizing the risk of mistakes.
- 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. CATIA MSoSA also provides a convenient report wizard for creating certification artifacts, including technical reports with graphs, diagrams, formulas, tables, and other relevant data.
- 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 2.2. Instead of relying on regular expressions, the Python companion code scans the UML file for model elements with non-empty `TestExpressions` and `TestValues` 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.

5. 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.

Two distinct approaches were presented. The first 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 defined 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).

Data Availability Statement: All datasets used to generate the certification artifacts and figures are publicly available. Additional materials supporting the findings are available from the corresponding author upon reasonable request. The analysis code and the SysML model will be released under an open-source license in a public repository. A persistent link will be provided upon request.

Acknowledgments: The authors thank Professor Giuseppe Di Gironimo and doctoral student 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:

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
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

References

1. Xie, J.; Harrison, E.D.; Mavris, D.N. Quantifying Impacts of Uncertainties on Certification-Driven Design. *J. Aircraft* **2024**, *61*, 1354–1377.
2. Schlickenmaier, H.; Voss, M.G.; Wilkinson, R.E. Certification Rules and Standards Review. *NASA Technical Report* **2019**.
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, 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, 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, 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, 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, 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, 2020; chapter AIAA Aviation 2020 forum, p. 3157, [https://arc.aiaa.org/doi/pdf/10.2514/6.2020-3157]. https://doi.org/10.2514/6.2020-3157.

 doi:10.20944/preprints202603.1390.v1

9. Xie, J.; Bendarkar, M.V.; Cai, Y.; Mavris, D.N. Evaluating "Critical-Engine-Inoperative" Conditions for Distributed Hybrid-Electric Aircraft. *J. Aircr.* **2025**, pp. 1–18.
10. 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, 2023, p. 3608.
11. 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.
12. 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. https://doi.org/https://doi.org/10.1016/j.trip.2025.101810.
13. Mirabella, C.; Tuccillo, M.; Vecchia, P.D. A model-based systems engineering digital certification framework for general aviation aircraft. *Aerotecnica Missili & Spazio* **2024**, pp. 1–16.
14. 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, 2019, p. 3576.
15. 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.
16. De Florio, F. *Airworthiness: An introduction to aircraft certification and operations*; Butterworth-Heinemann, 2016.
17. Cook, S.P. Digital airworthiness certification: Opportunities for unmanned aircraft systems. In Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2024, pp. 1432–1437.
18. Walgemoed, H. Flight envelope. *Introduction to Flight Test Engineering* **1995**.
19. 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.