NASA/CR-20250007045

Model-Based Systems Analysis and Engineering: Aircraft Data Hierarchy

Ranald M Engelbeck
The Boeing Company, Everett, Washington

Eduardo Ocampo
The Boeing Company, Huntington Beach, California

Joerg Gablonsky
The Boeing Company, Tukwila, Washington

Mingxuan Shi
The Boeing Company, Everett, Washington

Sean Wakayama
The Boeing Company, Huntington Beach, California

Alexander Carrere.
The Boeing Company, Huntsville, Alabama

Ronald Plybon
GE Aerospace, Cincinnati, Ohio

Melinda Refford
Collins Aerospace, Charlotte, North Carolina

Paul Mokotoff, Safa Bakhshi, Alex Kerlee, Gokcin Cinar, and Joaquim Martins
University of Michigan, Ann Arbor, Michigan

August 2025

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/CR-20250007045

Model-Based Systems Analysis and Engineering: Aircraft Data Hierarchy

Ranald M Engelbeck
The Boeing Company, Everett, Washington

Eduardo Ocampo
The Boeing Company, Huntington Beach, California

Joerg Gablonsky
The Boeing Company, Tukwila, Washington

Mingxuan Shi
The Boeing Company, Everett, Washington

Sean Wakayama
The Boeing Company, Huntington Beach, California

Alexander Carrere.
The Boeing Company, Huntsville, Alabama

Ronald Plybon
GE Aerospace, Cincinnati, Ohio

Melinda Refford
Collins Aerospace, Charlotte, North Carolina

Paul Mokotoff, Safa Bakhshi, Alex Kerlee, Gokcin Cinar, and Joaquim Martins
University of Michigan, Ann Arbor, Michigan

Prepared under Contract 80GRC023CA045

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

August 2025

Acknowledgments

This project and report reflect the combined efforts of the Boeing MBSA&E team. Funding for this effort was provided by the Sustainable Flight National Partnership Program, through the Advanced Air Transport Technology Project. The work was performed under NASA Contract 80GRC023CA045. Eric Hendricks was the NASA MBSA&E Program Manager. Ty Marien was the Contract Technical Representative. Irian Ordaz was the NASA sub-task lead for this work.

This work was sponsored by the Advanced Air Vehicles Program at the NASA Glenn Research Center.

Level of Review: This material has been technically reviewed by expert reviewer(s).

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

1.0 Abstract

The Aircraft Data Hierarchy (ADH) represents a transformative Model-based Systems Analysis & Engineering (MBSA&E) framework for standardized aerospace data exchange, specifically designed to address critical challenges in the aerospace industry's digital transformation of its multidisciplinary design processes. This comprehensive technical document outlines the development, structure, and prototype Python implementation of an open-source ADH as a solution that integrates systems analysis (MBSA) with systems engineering (MBSE) using a recursive Model-Based Systems-of-Systems Architecture (MSoSA) structure aligned with established standards.

The ADH facilitates efficient exchange of critical information—geometry definitions, disciplinary tool inputs/outputs, and engineering requirements—through a centralized, validatable data structure implemented using Pydantic v2, with support for serialization in JSON, YAML, and XML formats. This paper details the ADH's high-level architecture, including an example integration with NASA's MBSA&E framework, its hierarchical organization of aerospace data, and its ability to accommodate diverse aircraft configurations.

The use of a centralized authoritative data source like the ADH eliminates data duplication, enhances cross-disciplinary collaboration, improves traceability, and supports digital thread continuity. This ultimately enables more efficient and innovative aircraft design processes. With significant progress already achieved and continued collaboration with key aerospace stakeholders including NASA, Boeing, GE Aerospace, Collins Aerospace, and the University of Michigan, the ADH is poised to become a fundamental component of future aircraft design. Continued investment and industry involvement through the AIAA are needed to realize the full transformative potential of the ADH.

2.0 Table of Contents

3.0 List of Figures

4.0 Nomenclature

ADH: Aircraft Data Hierarchy - A standardized framework for storing, managing, and exchanging aircraft design and analysis data.
ADML: Aircraft Data Markup Language - A schema designed to standardize the representation of aircraft data, developed by researchers at Virginia Tech.
AGILE: Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts - A European Union project that included CPACS.
ANSI: American National Standards Institute - Organization that oversees the development of voluntary consensus standards in the United States.
AIAA: American Institute of Aeronautics and Astronautics - Professional society dedicated to the advancement of aerospace engineering.
API: Application Programming Interface - A set of rules that allows different software applications to communicate with each other.
ATA: Air Transport Association
ATA SPEC 2000: A set of e-business specifications, products, and services for the aviation industry developed by Air Transport Association.
CI/CD: Continuous Integration/Continuous Development – Aims to streamline and accelerate the software development lifecycle
CLIN: Contract Line Item Number - A specific task or deliverable defined in a government contract.
CPACS: Common Parametric Aircraft Configuration Schema - A data schema developed by DLR to facilitate the exchange of aircraft configuration data.
CRUD: CRUD refers to the fundamental operations of Create, Read, Update, and Delete, which are essential for managing persistent data in web services.
DLR: Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center) - The national center for aerospace, energy and transportation research of Germany.
EDP: Engineering Data Platform - A system for managing engineering data throughout the product lifecycle.
GenAI: Generative Artificial Intelligence - a subset of AI that focuses on creating new content, such as text, images, or music, by learning patterns from existing data and generating novel outputs that resemble the original data.
Git: Git is a distributed version control system for tracking changes in computer files and coordinating work on those files among multiple [1] people
Git Hub: GitHub is a web-based platform for version control and collaboration using Git
HDF5: Hierarchical Data Format version 5 - A file format designed to store and organize large amounts of numerical data.
ISO 10303: A standard for the computer-interpretable representation and exchange of product manufacturing information, also known as STEP.
JSON: JavaScript Object Notation - A lightweight data interchange format based on JavaScript object syntax.
MBE: Model-Based Engineering - An approach that uses models as the primary artifacts of the engineering process.
MBSA: Model-Based Systems Analysis - The application of modeling to support system analysis activities.
MBSA&E: Model-Based Systems Analysis and Engineering - The integration of model-based approaches for both systems analysis and engineering.
MBSE: Model-Based Systems Engineering - The formalized application of modeling to support system engineering processes.
MCP: Model Context Protocol - A protocol for enabling AI agents to interact with models.
MDS/MDF: Master Dimension Specification/Master Dimension File - Engineering specifications that define key dimensional aspects of an aircraft.
MIL-STD-881F: A military standard that establishes criteria for developing and presenting a Work Breakdown Structure (WBS).
MSoSA: Model-Based System-of-Systems Architecture - Guidelines for developing and describing complex Systems-of-Systems.
NASA: National Aeronautics and Space Administration - The United States government agency responsible for

the civilian space program and aerospace research.
NLP: Natural Language Processing - a field of artificial intelligence that focuses on the interaction between computers and humans through natural language, enabling machines to understand, interpret, and generate human language.
OpenAPI: A specification for machine-readable interface files for describing, producing, consuming, and documenting RESTful web services.
OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization developed by NASA.
PIDO: Process Integration and Design Optimization
PLM: Product Lifecycle Management - The process of managing the entire lifecycle of a product from inception through design, manufacture, service, and disposal.
Pydantic: A data validation and settings management library using Python type annotations.
RESTful: Representational State Transfer - An architectural style for designing networked applications.
SAWE RP A-8: Society of Allied Weight Engineers Recommended Practice A-8 - A standard for weight and balance reporting for aircraft.
STEP: Standard for the Exchange of Product model data - Formally known as ISO 10303.
SonarCube: A platform for continuous inspection of code quality to perform automatic reviews with static analysis of code.
URL: A URL (Uniform Resource Locator) is essentially a web address that specifies the location of a resource on the internet and the mechanism for retrieving it.
VPI: Virginia Tech, officially the Virginia Polytechnic Institute and State University, is a public land-grant research university in Blacksburg, Virginia.
WBS: Work Breakdown Structure - A hierarchical decomposition of project work into smaller, more manageable components.
XML: Extensible Markup Language - A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
YAML: YAML Ain't Markup Language - A human-readable data serialization standard often used for configuration files.

5.0 Executive Summary

The Aircraft Data Hierarchy (ADH) represents a critical advancement in the standardization of aerospace data exchange, specifically designed to meet the complex requirements of aerospace vehicle design studies. This prototype data definition standard addresses fundamental quality gaps in the aerospace industry's current data exchange infrastructure by introducing a comprehensive, hierarchical framework that seamlessly integrates both systems analysis and systems engineering perspectives.

5.1 Key Features and Structure

The ADH's distinctive value proposition lies in its recursive structure based on Model-Based Systems-of-Systems Architecture (MSoSA) guidelines and its alignment with established industry standards, including MIL-STD-881F, SAWE RP A-8, and ANSI/AIAA-S-119-2011. The framework enables efficient exchange of critical information—geometry definitions, disciplinary tool inputs/outputs, and engineering requirements—through a common, validatable data structure that bridges previously siloed processes.

Implemented using modern programming methodologies, including schema definition and validation through Pydantic v2, shown in Figure 1, enables the ADH to support multiple persistence formats (JSON, YAML, and XML). This flexibility ensures compatibility with diverse technological environments and facilitates seamless integration into existing workflows and tools.

```python
class LiftingSurface(CommonBaseModel):
    tip_chord: Optional[float] = Field(None, ge=0, description="Tip chord")
    outboard_panel_semi_span: Optional[float] = Field(None, ge=0, description="Outboard panel semi-span")
    exposed_panel_semi_span: Optional[float] = Field(None, ge=0, description="Exposed panel semi-span from side-of-body")
    total_panel_semi_span: Optional[float] = Field(None, ge=0, description="Theoretical panel semi-span from centerline")
    breakpoint_chord: Optional[float] = Field(None, ge=0, description="Chord at break point")
    root_chord: Optional[float] = Field(None, ge=0, description="Root chord")
    inboard_panel_sweep: Optional[float] = Field(None, description="Inboard panel sweep angle")
    outboard_panel_sweep: Optional[float] = Field(None, description="Outboard panel sweep angle")
    reference_chord_fraction: Optional[float] = Field(0.25, gt=0, lt=1, description="Reference chord fraction for inboard and outboard sweep angles")
    twist_angle: Optional[float] = Field(None, description="Twist angle, negative leading edge rotated down")
    inboard_panel_dihedral: Optional[float] = Field(None, description="Inboard panel dihedral angle")
    outboard_panel_dihedral: Optional[float] = Field(None, description="Outboard panel dihedral angle")
    planform_type: Optional[PlanformType] = Field(None, description="Planform type: 1.0 Straight tapered, 2.0 - Double Delta, 3.0 - Cranked")
    shock_impengement_area: Optional[float] = Field(None, ge=0, description="Fuselage portion covered by shock zone emanating from root of horizontal")
    extended_shock_impengement_area: Optional[float] = Field(None, ge=0, description="Extended Fuselage portion covered by shock zone emanating from root of horizontal")
    distance_between_CG_and_centroid: Optional[float] = Field(None, description="Longitudinal Distance between CG and centroid of stabilizer")
    vertical_panel_exposed_root_chord: Optional[float] = Field(None, ge=0, description="Exposed Vertical Panel Area of Wing exposed root chord")
    vertical_panel_not_influenced_by_wing: Optional[float] = Field(None, ge=0, description="Exposed Vertical Panel Area not influenced by Wing or Horizontal")
    horizontal_panel_exposed_root_chord: Optional[float] = Field(None, ge=0, description="Exposed Vertical Panel Area of Horizontal exposed root chord")

    @field_validator('tip_chord', 'outboard_panel_semi_span', 'exposed_panel_semi_span', 'total_panel_semi_span', 'breakpoint_chord', 'root_chord', 'shock_impengement_area',
        'extended_shock_impengement_area', 'vertical_panel_exposed_root_chord', 'vertical_panel_not_influenced_by_wing', 'horizontal_panel_exposed_root_chord', mode='before')
    def check_non_negative(cls, v):
        if v is not None and v < 0:
            raise ValueError("Values must be non-negative")
        return v

    @field_validator('reference_chord_fraction', mode='before')
    def check_chstat_range(cls, v):
        if v is not None and (v <= 0 or v >= 1):
            raise ValueError("Reference chord fraction must be between 0 and 1")
        return v
```

Figure 1- Example Parametric Lifting Surface Pydantic Class

5.2 Benefits

The ADH offers several significant benefits:

1. **Standardization and Consistency**: Provides a uniform framework for data representation, reducing ambiguities and enhancing cross-disciplinary collaboration.
2. **Interoperability**: Promotes compatibility across different systems and platforms, minimizing proprietary dependencies and enabling comprehensive, multi-disciplinary design and analysis.
3. **Flexibility and Adaptability**: Accommodates diverse aircraft configurations and evolving design methodologies, ensuring the framework remains relevant and useful as aerospace technology advances.
4. **Enhanced Efficiency and Reliability**: Centralizes data management, eliminating duplication, reducing manual data translation, and improving data integrity and traceability.
5. **Seamless Integration and Collaboration**: Facilitates integration with existing tools and promotes collaboration among project teams and stakeholders, enhancing communication and reducing design conflicts.

5.3 Current Progress

Significant progress has been made in the development of the ADH, including:

- **Foundational Development**: Establishing the essential infrastructure and coding standards.
- **Requirements Gathering**: Engaging with industry stakeholders to finalize data requirements and develop detailed use cases.
- **Architectural Decisions**: Aligning the ADH with MSoSA guidelines and industry standards.
- **Prototyping and Integration**: Developing a prototype schema and Python package, and conducting initial integration tests with NASA's MBSA&E framework.
- **Demonstrations**: Showcasing the ADH's capabilities in real-world scenarios, including integration with propulsion system design workflows and bidirectional data exchange with MagicDraw.

5.4 Industry Involvement and Collaboration

The success of the ADH depends on its adoption and support by the broader aerospace community. The development has actively engaged key stakeholders, including NASA, Boeing, GE Aerospace, Collins Aerospace, and the University of Michigan. Future involvement of additional universities, defense organizations, and aircraft manufacturers will further enhance the framework's versatility and promote widespread adoption.

5.5 Next Steps

To fully realize the transformative potential of the ADH, the following next steps are recommended:

1. **Detailing Data Objects**: Define the specific parameters and properties of individual data objects to ensure comprehensive coverage of all relevant aspects of aircraft design as illustrated in Figure 1.
2. **Enhanced Integration**: Develop standardized APIs and web services to facilitate easier integration with diverse systems and tools.
3. **Advanced Capabilities**: Incorporate features such as uncertainty quantification, system-level rollups, and AI integration to enhance the utility and impact of the ADH.
4. **Comprehensive Documentation**: Provide detailed documentation, tutorials, and training resources to support adoption and effective use of the ADH.
5. **Open Source Collaboration**: Publish the ADH on a public GitHub repository to invite broader industry participation and leverage collective expertise.

5.6 Conclusion

The Aircraft Data Hierarchy represents a transformative framework for aerospace data standardization, offering a comprehensive solution to long-standing challenges in data integration and cross-disciplinary collaboration. By providing the standardization, interoperability, flexibility, and efficiency necessary for modern aerospace design, the ADH enables more efficient, effective, and innovative approaches to aircraft design and analysis. The progress achieved to date, combined with continued investment and industry collaboration, positions the ADH to become a fundamental component of future aircraft design processes. This strategic initiative will drive innovation, enhance efficiency, and maintain NASA's leadership in aerospace technology development, ultimately advancing the aerospace industry as a whole.

6.0 Introduction: The Critical Need for an MBSE Aligned Standard Data Repository

The aerospace industry has long struggled with the inefficiencies inherent in manual data transfer between different engineering disciplines. This is one of the major reasons for the aerospace industries ongoing Digital Transformation. These processes are time-consuming, error-prone, and create significant barriers to innovation.

The ADH addresses these challenges by providing a centralized, standardized framework for data exchange that facilitates seamless communication and collaboration among different engineering teams and tools. The need for and efficiencies of a centralized source of authoritative data have been known within Boeing and the aerospace industry for decades. DLR's Common (CPACS) and VPI's Aircraft Data Modeling Language (ADML) being two example implementations. Both papers (References 15 and 16) document the benefits and efficiencies of integrating engineering tools to a centralized data source as illustrated in Figure 2 taken from Reference 15.

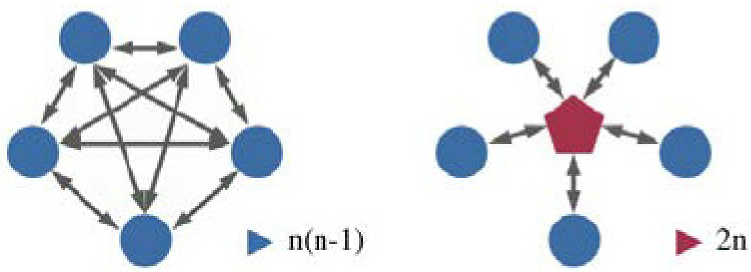

Figure 2- Centralized Data Repositories have fewer interfaces to maintain and separate data integration from changes in process or tools

There are two main advantages of centralized data sources: a reduction in the number of interfaces to maintain and the separation of data from processes. Tool-to-tool integration often requires rework when the problem statement or tools change. By having a centralized data repository or hub, such as the ADH, a library of tools can be integrated with the data repository. This allows any process to be executed simply by changing the order in which the tools are used.

The Aircraft Data Hierarchy (ADH) implementation addresses an additional challenge within the aerospace engineering community: the absence of a standardized MBSE aligned, interoperable framework for authoritative aircraft data exchange. Figure 3 shows that the ADH prototype as the central data hub/repository for the MBSA&E framework which is built on the NASA open-source packages Aviary and OpenMDAO (References 17 and 18). The ADH facilitates more efficient and effective conceptual and preliminary design of aircraft through improved collaboration/interoperatbility between NASA SFNP projects and the MBSA&E project in addition to the other engineering organizations tools.

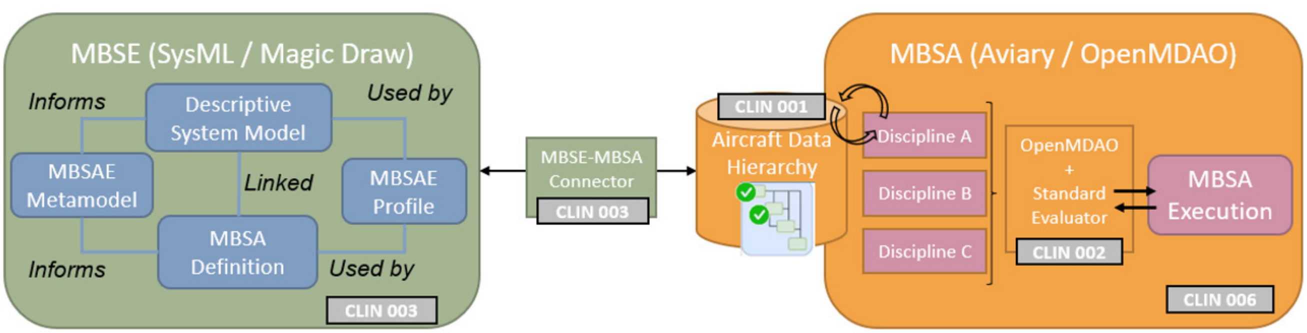

Figure 3 – The MBSA&E Concept Framework Architecture utilizes the Aircraft Data Hierarchy (ADH) as a common source to enable Aircraft Data exchange between MBSE and MBSA.

For Model-Based Systems Engineers at NASA, this represents a significant step forward in addressing fundamental challenges in current data exchange practices; manual or automated. The ADH offers a comprehensive solution that improves standardization, interoperability, flexibility, and efficiency across the industry's aircraft design and analysis processes.

7.0 Current Challenges in Aircraft Data Management

NASA's Aviary tool deserves recognition for its structured data hierarchy, which has been instrumental in organizing and managing data. However, there are opportunities to improve the current data exchange infrastructure to further enhance the efficiency and effectiveness of aircraft design and analysis processes beyond Aviary. These opportunities are particularly relevant for Model-Based Systems Engineers and underscore the urgent need for a more comprehensive, global solution.

The existing systems demonstrate a pronounced bias toward systems analysis rather than providing a balanced approach that integrates various engineering disciplines. This imbalance creates significant challenges for systems engineers who must work across disciplinary boundaries. The siloed nature of these systems inhibits the holistic approach essential for successful aerospace design, forcing engineers to develop custom, ad-hoc solutions for cross-disciplinary data exchange.

Furthermore, current data hierarchies are predominantly tool-oriented, focusing on the specific requirements of individual software applications rather than on the comprehensive representation of aircraft data throughout the design process. This tool-centric approach creates dependencies on specific applications and restricts the flexibility needed to adapt to evolving design methodologies and technologies. Systems engineers often find themselves constrained by the limitations of their tools, unable to implement more efficient or innovative approaches due to the lack of a centralized authoritative data source between the engineering disciplines.

Perhaps most critically, there exists a notable absence of common data standards between Model-Based Systems Engineering (MBSE) and Model-Based Systems Analysis (MBSA) models. This disconnect creates substantial barriers to seamless information exchange and integration, resulting in redundant data entry, inconsistencies across models, and challenges in maintaining design coherence throughout the development lifecycle. The lack of standardization requires extensive manual intervention to translate between different data representations, consuming valuable engineering resources and introducing potential points of failure.
These challenges collectively contribute to prolonged development timelines, increased project costs, and limitations in design innovation—all issues that directly impact NASA's ability to maintain leadership in aerospace technology development. For Model-Based Systems Engineers, these constraints represent significant obstacles to implementing comprehensive, efficient design methodologies that leverage the full potential of modern computational tools and collaborative engineering approaches.

8.0 Survey of Existing Solutions

In developing the ADH, our team conducted an exhaustive evaluation of existing data formats and frameworks to identify potential solutions that could meet NASA's sophisticated requirements. This comprehensive survey revealed significant gaps in current approaches while also identifying valuable components that could inform the development of a more comprehensive solution.

The Common Parametric Aircraft Configuration Schema (CPACS) discused in reference 15 and illustrated in Figure 4, developed by the German Aerospace Center (DLR) in 2005, emerged as one of the most mature solutions currently available. CPACS has demonstrated success in exchanging information at the conceptual design level and has been implemented in various international projects, including the European Union's AGILE project discussed in reference 14 and illustrated in Figure 5. Its structured format for representing aircraft configurations offers valuable insights for the development of the ADH.

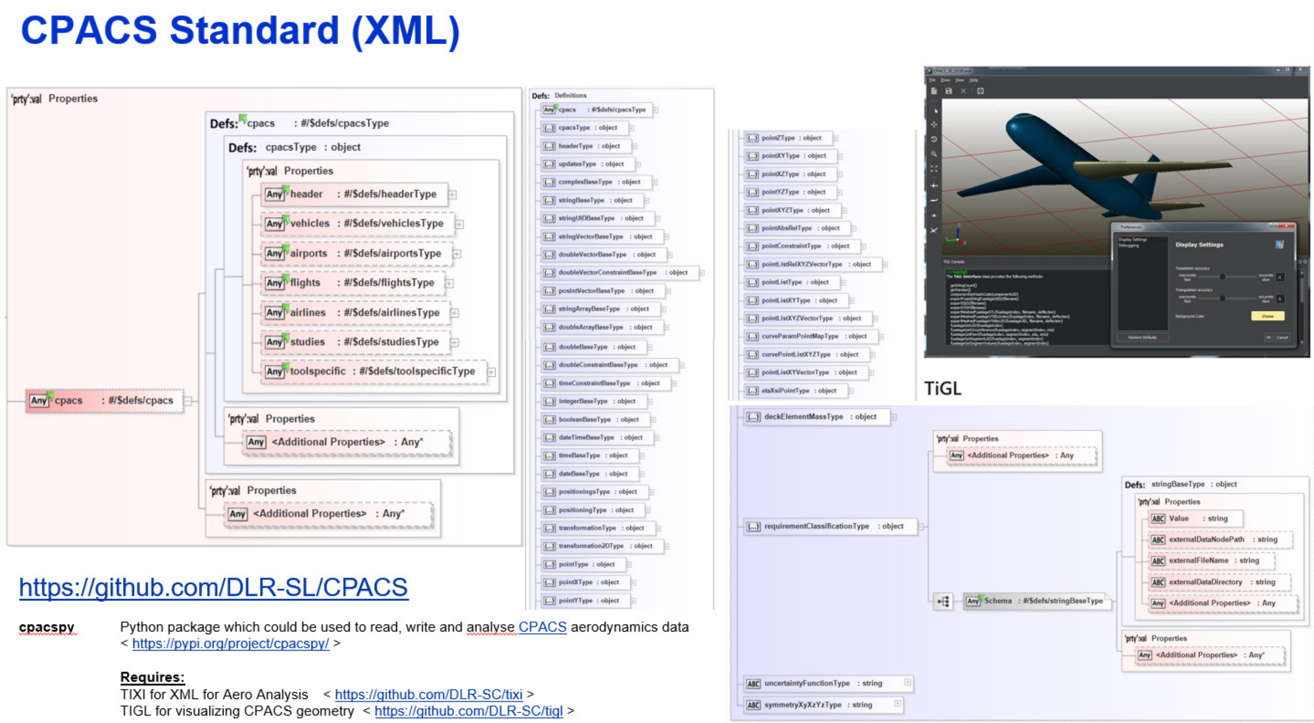

Figure 4- Illustration of the CPACS Standard (XML) and TiGL visualization tool, showcasing the structured data hierarchy and geometry visualization capabilities for aerospace design.

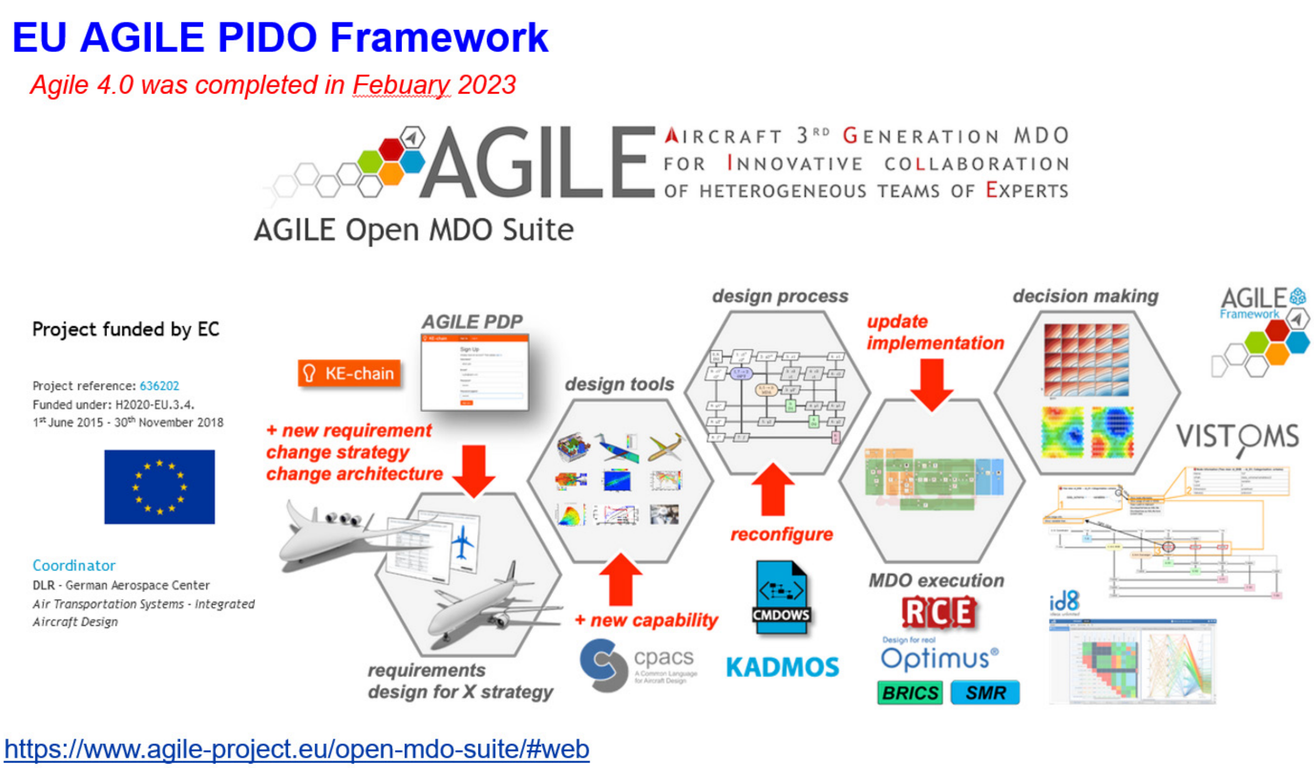

Figure 5- Overview of the EU AGILE PIDO Framework, illustrating the multidisciplinary design optimization (MDO) suite and its integration with various design tools and processes for innovative collaboration.

However, while CPACS provides a solid foundation, it falls short in several critical areas necessary for NASA's comprehensive needs. Specifically, CPACS lacks robust integration with Model-Based Systems Engineering (MBSE) methodologies and has limited support for the full range of disciplinary analyses required in advanced aerospace design. For Model-Based Systems Engineers at NASA, these limitations would constrain the application of CPACS to only a subset of the design analysis process, necessitating additional tools and frameworks to address the full spectrum of systems engineering activities.

The Aircraft Data Markup Language (ADML) is discussed in reference 16. We converted the ADML schema into Pydantic classes with their signatures shown in Figure 6 to show what Pydantic data classes would be available. ADML developed by researchers at Virginia Tech, offers another specialized framework for aircraft data representation. However, its limited support for MBSE, limited adoption within the broader aerospace community and its focus on specific aircraft types restrict its applicability as a comprehensive solution for NASA's diverse design activities.

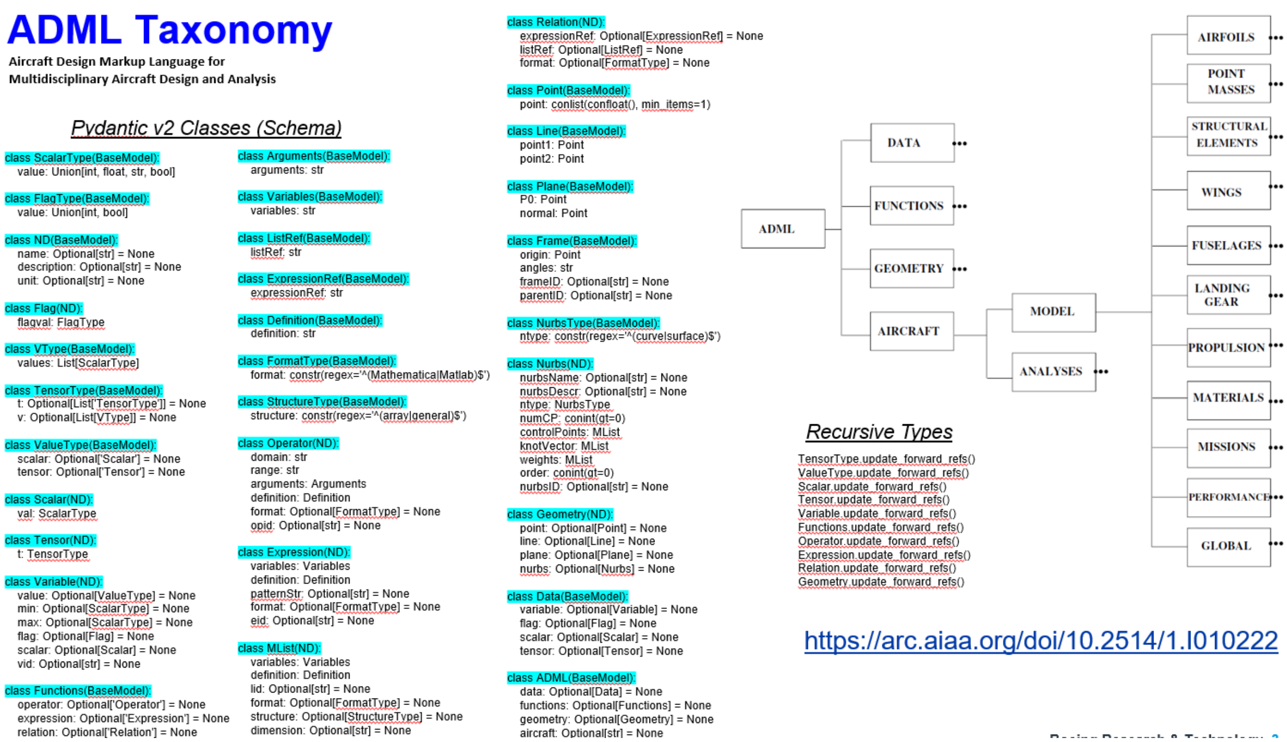

Figure 6- ADML Taxonomy showcasing the Pydantic v2 classes (schema) and the hierarchical structure for multidisciplinary aircraft design and analysis.

Details of our assessment of CPACS, ADML and the ADH are summarized in Figures 7 and 8.

Aspect	ADML	CPACS	ADH
Standardization	ADML provides a standardized schema for representing aircraft data, focusing on detailed low-level schemata. This standardization ensures consistency in data representation across different tools and disciplines.	CPACS offers a standardized data model for aircraft, rotorcraft, and spacecraft design, with a focus on high-level constructs and flexibility through a middle-out approach. It is widely adopted in the aerospace community, promoting consistency and interoperability.	ADH aims to provide a comprehensive, standardized framework for aerospace data exchange, integrating systems analysis and systems engineering perspectives. It aligns to established standards like MIL-STD-881F and ANSI/AIAA-S-119-2011, ensuring consistency and reliability.
Interoperability	ADML supports interoperability by providing a common language for data communication, facilitating seamless data exchange between different tools and platforms.	CPACS enhances interoperability by serving as a central data model, reducing the number of data interfaces and ensuring consistent information exchange across various tools and disciplines.	ADH promotes interoperability through vendor-neutral standards and open data formats (JSON, YAML, XML), enabling seamless integration with diverse systems and tools. It supports comprehensive MDAO workflows and integration with OpenMDAO.
Flexibility	ADML is designed to be flexible, accommodating various aircraft configurations and supporting both high-level and detailed representations. Its bottom-up approach allows for easy extension and adaptation.	CPACS provides flexibility through its middle-out approach, combining top-down and bottom-up methodologies. It supports detailed parametrization for various components and allows for tool-specific schemas, ensuring adaptability to different design needs.	ADH offers inherent flexibility, supporting different levels of fidelity and detail throughout the design process. Its recursive MSoSA structure and support for multiple data formats ensure adaptability to evolving design methodologies and technologies.
Efficiency	ADML enhances efficiency by minimizing data duplication and ensuring consistent data representation. Its detailed low-level schemata facilitate efficient data management and processing.	CPACS improves efficiency by reducing the number of data interfaces and ensuring consistent information exchange. Its hierarchical structure and detailed parametrization support efficient data management and analysis.	ADH ensures efficient data management through automation of common tasks (e.g., unit conversion, data validation) and comprehensive validation capabilities. Its integration with OpenMDAO and support for advanced analytics enhance overall design efficiency.
Traceability	ADML provides provenance capability, allowing for detailed traceability of data origin and history. This supports rigorous systems engineering processes and enhances accountability.	CPACS does not explicitly mention provenance capability, but its structured data model supports traceability through consistent data representation and hierarchical organization.	ADH offers comprehensive traceability and documentation of design decisions, supporting rigorous systems engineering processes and ensuring accountability throughout the design lifecycle.
Integration with MBSE	ADML supports MBSE by providing a detailed, standardized schema for representing aircraft data, facilitating integration with systems engineering tools and processes.	CPACS supports MBSE by serving as a central data model for aircraft design, promoting consistent information exchange and integration across different disciplines and tools.	ADH is specifically designed to integrate systems analysis and systems engineering perspectives, providing a unified framework for MBSE. Its hierarchical MSoSA structure and comprehensive data representation support seamless integration with MBSE methodologies.
Support for MDAO	ADML supports MDAO by providing a common language for data communication and facilitating seamless data exchange between different tools and platforms.	CPACS enhances MDAO by serving as a central data model, reducing the number of data interfaces and ensuring consistent information exchange across various tools and disciplines.	ADH supports comprehensive MDAO workflows, enhancing cross-disciplinary collaboration and integration. Its integration with OpenMDAO and support for advanced analytics enable efficient multidisciplinary design optimization.
Adoption and Community Support	ADML is developed by researchers at Virginia Tech and has limited adoption within the broader aerospace community.	CPACS is widely adopted in the aerospace community, with extensive use in DLR projects and international collaborations. Its open-source development model and community support promote widespread adoption.	ADH is developed with collaboration from key aerospace stakeholders, including NASA, Boeing, GE Aerospace, and Collins Aerospace. Its open-source development model and comprehensive stakeholder engagement ensure broad adoption and support.

Figure 7- Comparison table of ADML, CPACS, and ADH, highlighting key aspects such as standardization, interoperability, flexibility, efficiency, traceability, integration with MBSE, support for MDAO, and adoption and community support.

Feature/Capability	ADML	CPACS	ADH
Modular Development	Logical decomposition of XML elements into subsets, enabling reusability and extensibility.	Follows a top-down approach with high-level constructs, supports middle-out approach for flexibility.	Hierarchical organization through MSoSA, modular and recursive structure.
Object-Oriented Approach	Hierarchical-type system resembling object-oriented programming, facilitating reusability and extensibility.	Not explicitly mentioned, but supports hierarchical data structures and object-oriented principles through XML.	Uses Pydantic for schema definition and validation, supports object-oriented principles.
Provenance Capability	Describes the origin or history of the associated data, including author, date, etc.	Not explicitly mentioned.	Comprehensive traceability and documentation of design decisions.
Low-Level Schemata	Common components such as data, functions, and basic geometry.	High-level constructs with less emphasis on low-level data elements, but includes detailed parametrization for components like engine nacelles and	Detailed specifications for individual data objects, comprehensive data representation.
Data Schema	Represents the simplest form of data, including variables, tensors, and scalars.	Uses pointLists and vectors for data representation, supports compact vector and array types for large datasets.	Supports multiple data formats (JSON, YAML, XML), robust data validation with Pydantic.
Mathematical Representation	Supports Mathematica or MATLAB syntax for representing mathematical objects, avoiding verbose formats like MathML.	Supports MathML for mathematical modeling, includes flexible aerodynamic performance maps.	Supports advanced analytics and AI integration, including uncertainty quantification.
Basic Geometry	Supports NURBS-based geometry model to represent curves and surfaces.	Uses pointLists for geometry representation, supports body-fitted coordinates for wing component segments.	Comprehensive geometry definitions, including support for OpenMDAO integration.
High-Level Aircraft Design Constructs	Includes elements for modeling aircraft geometry, wings, fuselages, landing gears, propulsion, materials, missions, and performance.	Includes elements for aircraft, rotorcraft, engines, profiles, structural elements, materials, fuels, missions, and flights.	Hierarchical organization through MSoSA, detailed WBS integration.
Airfoil Definition	Supports parametric, NURBS-based, VT-CST-based geometry definitions, or references to external definitions.	Defined as pointLists, supports flexible aerodynamic performance maps.	Comprehensive and flexible airfoil definitions, supporting various configurations.
Point Masses	Simple description of location, mass, and moment of inertia tensor for internal components.	Not explicitly mentioned.	Detailed mass properties and rollup capabilities for system-level assessments.
Structural Elements	Definitions for isotropic and composite materials, supporting primary structural members like spars, ribs, and skin panels.	Supports structural elements with detailed parametrization for wing internal structures.	Comprehensive structural definitions, including materials and performance metrics.
Wings	Defined by planform, structure, control effectors, and composite wing boxes, supporting various configurations.	Defined with high-level constructs, supports body-fitted coordinates and flexible aerodynamic performance maps.	Detailed wing definitions, supporting various configurations and performance analyses.
Fuselages	Parametric geometry definition using VT-CST.	Defined with high-level constructs, supports detailed parametrization.	Comprehensive fuselage definitions, supporting various configurations.
Landing Gears	Supports tricycle, quadricycle, and multibogey configurations.	Defined with high-level constructs.	Detailed landing gear definitions, supporting various configurations.
Propulsion	Includes engines, cowls, ramps, and EEWSs, with VT-CST parametric definition.	Includes engines and engine pylons, supports detailed parametrization of engine nacelles.	Comprehensive propulsion system definitions, supporting various configurations and performance analyses.
Materials	Catalog element for isotropic and orthotropic materials.	Includes materials and fuels, supports detailed parametrization.	Detailed material definitions, supporting various analyses and performance metrics.
Missions	List of mission segments for flight performance analysis and optimization.	Includes missions and airports, supports detailed and flexible mission definitions.	Comprehensive mission definitions, supporting various analyses and performance metrics.
Performance	Encapsulates high-level information regarding aircraft behavior or limitations.	Supports detailed aerodynamic performance maps and operational limits.	Detailed performance metrics and rollup capabilities for system-level assessments.
Comparison with CPACS	Follows a bottom-up approach with detailed low-level schemata, more efficient and expressive at the data level.	Follows a top-down approach, high-level constructs, supports middle-out approach for flexibility, detailed parametrization for various components.	Comprehensive, hierarchical framework, integrating systems analysis and engineering perspectives.
Integration with Tools	Demonstrated integration with design optimization in C++ (DOC) project and use of XSD for converting XML schema to C++ classes.	Integrated with various DLR projects and tools like TIGL, supports tool-specific schemas and decentralized workflows.	Seamless integration with OpenMDAO, supports multiple data formats and advanced analytics.
Proof of Concept	Encoding of the entire Convair B58 aircraft, demonstrating ADML's capability to represent complex aircraft models.	Widely adopted in DLR projects, supports various aircraft configurations, demonstrated in projects like AGILE and IDEaliSM.	Demonstrated capabilities in real-world scenarios, including integration with propulsion system design and analysis workflows.
MDAO Workflow Design	Not explicitly mentioned.	MDAx provides an intuitive workflow modeling environment using XDSM format with additional design rules.	Supports comprehensive MDAO workflows, enhancing cross-disciplinary collaboration and integration.

Figure 8- Detailed comparison table of ADML, CPACS, and ADH, highlighting their features and capabilities across various aspects of aerospace data management and design.

Our survey also encompassed a diverse range of data formats, including HDF5, XML, JSON, and YAML. Our evaluation of their strengths and limitations for aerospace data representation is shown in Figure 9. We also considered the needs of software developers by including each technology's exposure to IntelliSence in programming Integrated Development Environments (IDE). While each offers specific advantages in terms of human readability, parsing efficiency, and integration capabilities, none provides a comprehensive solution for the complex, multidisciplinary nature of aerospace design data.

Data Access and Persistence of Data in files

Feature / Format	HDF5	JSON	YAML	XML
Data Structure	Hierarchical (groups and datasets)	Hierarchical key-value pairs	Hierarchical key-value pairs	Hierarchical elements and attributes
Data Types	Preserves most Python data types, including NumPy arrays	Limited data types, may require type conversions	Limited data types, may require type conversions	Limited data types, may require type conversions
Indexing/Querying	Efficient indexing and querying within the file	Limited querying capabilities	Limited querying capabilities	Limited querying capabilities
Human Readability	Requires specialized tools	Human-readable	Human-readable	Less human-readable
Compatibility	Widely used in scientific computing	Widely supported across languages	Supported in many languages	Supported in many languages
Schema Validation	Supports datatypes and attributes for validation	Schema validation options exist	Schema validation options exist	Strict validation with XSD schemas
Recommended for	Large datasets, numerical data, performance-critical	Widely compatible data exchange, human readability	Human-readable configuration files, data exchange	Structured data with strict validation requirements

Comparative Assessment of Data Structures

Format	Ease of Use	Human Visualization	Computational Efficiency	Programming Ease	Scalability	Robustness	Data Validation	Type Checking	Units	Axis System Integrity	Ranking
HDF5	Medium	Low	High	Medium	High	High	Medium	Medium	Medium	Medium	2
Pydantic Classes	High	Medium	Medium	High	Medium	High	High	High	Medium	Medium	5
Nested Python Dictionaries	High	Medium	Medium	High	Medium	Medium	Low	Low	Low	Low	-2
Pandas DataFrames	High	High	Medium	High	Medium	Medium	Medium	Medium	Medium	Low	2
XML	Medium	Medium	Low	Medium	Medium	High	Medium	Medium	Low	Low	-2
JSON	Medium	Medium	Medium	Medium	Medium	Medium	Low	Low	Low	Low	-4
YAML	Medium	Medium	Medium	Medium	Medium	Medium	Low	Low	Low	Low	-4

Figure 9- Comparative assessment of data structures and formats for data access and persistence, highlighting their features, capabilities, and suitability for aerospace data management.

Industry standards such as ISO 10303 (STEP) were carefully evaluated for their applicability in defining a comprehensive data hierarchy. While STEP provides valuable standardization for product manufacturing data, its focus on downstream manufacturing rather than early-stage conceptual and preliminary design limits its utility for NASA's specific objectives. The extensive overhead associated with STEP implementation also presents challenges for rapid iteration and agile design methodologies increasingly adopted in aerospace engineering.

Summary

ADML:

- **Strengths**: Detailed low-level schemata, flexibility, efficiency, provenance capability, supports MBSE and MDAO.
- **Weaknesses**: Limited adoption within the broader aerospace community.

CPACS:

- **Strengths**: Standardization, interoperability, flexibility through middle-out approach, efficiency, widely adopted, supports MBSE and MDAO.
- **Weaknesses**: Less emphasis on detailed low-level data elements, no explicit provenance capability.

ADH:

- **Strengths**: Comprehensive standardization, interoperability, flexibility, efficiency, traceability, integration with MBSE and MDAO, broad adoption and support.
- **Weaknesses**: Still in development, requires continued investment to realize full potential.

From the perspective of a Model-Based Systems Engineer, ADH offers the most comprehensive and integrated framework for MBSE, addressing critical challenges in data management, standardization, and interoperability. CPACS provides a well-established, flexible data model with broad adoption, while ADML offers detailed low-level schemata and flexibility but has limited adoption.

Through this evaluation, it became evident that while existing solutions offer valuable components and insights, none provides the comprehensive, integrated framework necessary to meet NASA's specific requirements for Model-Based Systems Engineering (MBSE). Simply adding an MBSE data branch to an existing data hierarchy would not provide the necessary data context to support MBSE. This gap underscores the critical need for the development of the Aircraft Data Hierarchy—a solution specifically designed to address these limitations and provide a robust, flexible framework for aerospace data exchange.

9.0 Requirements and Objectives

The development of the Aircraft Data Hierarchy is guided by four fundamental objectives that address the critical needs of Model-Based Systems Engineers at NASA: standardization, interoperability, flexibility, and efficiency. These objectives establish the foundation for a comprehensive framework that will transform how aircraft data is managed, exchanged, and utilized throughout the design process.

To be an enduring standard, the ADH also needs to embrace the latest software development standards and emerging technolgies: Notabily Open Source DevOps, and Artificial Intelligence.

9.1 Standardization

Aligning with standards represents the cornerstone of the ADH initiative. For Model-Based Systems Engineers, a standardized data hierarchy provides consistency and clarity in data representation, eliminating ambiguities and reducing the potential for misinterpretation. The ADH aims to develop a standardized yet adaptable data hierarchy that can be consistently applied across different topologies of aircraft design projects, from conventional fixed-wing aircraft to advanced rotorcraft and novel configurations.

This alignment ensures uniformity in data representation and management, facilitating more effective collaboration and communication among project teams. By establishing common terminology, data structures, and relationships, the ADH creates a shared "language" for aircraft design and analysis, reducing the cognitive overhead associated with translating between different data representations and improving the efficiency of cross-disciplinary collaboration.

9.2 Interoperability

Interoperability extends beyond basic data exchange to enable seamless integration of different systems, tools, and processes. The ADH promotes interoperability and compatibility across different systems and platforms by adopting vendor-neutral standards and open source data formats. This approach minimizes proprietary dependencies and ensures that the data hierarchy can function effectively in diverse technological environments. For NASA's Model-Based Systems Engineers, interoperability means the ability to integrate multiple disciplinary analysis tools seamlessly, enabling comprehensive, multi-disciplinary aircraft design and analysis without the manual data translation that currently consumes significant engineering resources. Alignment of this capability with Systems Engineering best practices is essential for implementing model-based systems engineering methodologies that rely on consistent data representation across different modeling and analysis environments.

Opening the development of the ADH as an open-source project invites the participation of industry partners, enhances the quality of the ADH, and fosters collaboration across industry and academia.

9.3 Flexibility

The aerospace industry continuously evolves, with new technologies, methodologies, and requirements emerging regularly. The ADH must accommodate this evolution by providing a flexible framework that can adapt to changing needs without requiring fundamental restructuring. Creating a flexible hierarchy that can incorporate unforseen aircraft systems and components as they are developed ensures that the data hierarchy remains relevant and useful as aerospace technology advances.

This flexibility extends to supporting various levels of fidelity and detail throughout the design process. The ADH must accommodate both high-level, conceptual representations and detailed, component-level specifications, enabling engineers to transition seamlessly between different levels of abstraction as the design matures. This capability is particularly valuable for Model-Based Systems Engineers, who must maintain consistency and traceability throughout the design lifecycle in the form of a Digital Thread.

9.4 Efficiency

Efficiency in data management is critical for supporting the complex analyses and simulations that characterize modern aerospace design. The ADH ensures efficient storage and configuration management of aircraft data, supporting rapid data access and processing even for large, complex datasets. This efficiency extends to minimizing the manual effort required for data management, allowing engineers to focus on value-added design and analysis activities rather than transforming data tasks.

By providing automation for common data management tasks, such as unit conversion, coordinate system transformation, and data validation, the ADH significantly reduces the time and effort required to prepare and maintain design data. This automation not only improves efficiency but also reduces the potential for errors and inconsistencies, enhancing the reliability, quality and accuracy of aircraft design and analysis outcomes.

These requirements and objectives establish a comprehensive framework for the development of the ADH, ensuring that it meets industry's current needs while providing the flexibility and extensibility necessary for future advancements in aircraft design and analysis methodologies. For Model-Based Systems Engineers, the ADH represents a significant advancement in access to the tools and frameworks available for implementing comprehensive, integrated design methodologies that leverage the full potential of modern computational capabilities.

10.0 ADH Structure and Philosophy

The Aircraft Data Hierarchy (ADH) is built upon a comprehensive philosophical foundation that guides its design, implementation, and application. The structure of the ADH is aligned with important standards and the MSoSA guidelines to logically organize its complex hierarchical data representation. It follows the principle of similar treatment, ensuring consistency and coherence throughout the data hierarchy.

10.1 Vision for Impact

The ultimate vision for the Aircraft Data Hierarchy extends beyond merely addressing current challenges to fundamentally transforming how aerospace engineering is conducted at NASA and throughout the industry. For Model-Based Systems Engineers at NASA, the ADH represents not just an incremental improvement in data management, but a paradigm shift in Model Based Systems Engineering (MBSE) that will enable more efficient, effective, and innovative approaches to aircraft design.

The ADH aims to establish itself as a fundamental component of modern aircraft design and analysis processes, driving innovation and excellence in aerospace engineering. By providing a standards-aligned, interoperable framework for aircraft data, the ADH will eliminate the barriers that currently impede cross-disciplinary collaboration and integration, enabling truly holistic approaches to aircraft design that consider multiple disciplines and objectives simultaneously.

This transformation will accelerate the design process, reducing the time and cost associated with aerospace development while enhancing the quality and reliability of the resulting designs. By automating routine data management tasks and ensuring consistency across different analyses and tools, the ADH will free engineers to focus on creative problem-solving and innovation rather than administrative data handling.

Furthermore, the ADH will enable more comprehensive exploration of the design space, supporting parametric studies, optimization, and trade-off analyses that would be prohibitively time-consuming with current approaches. This enhanced capability will lead to more efficient, capable, and innovative aircraft designs that better meet NASA's mission objectives and advance aerospace technology.

10.2 Comprehensive Data Representation

The ADH is designed to include all necessary data to meet the requirements for aircraft design and validation processes, creating a comprehensive, integrated representation of the aircraft system. This completeness ensures that the ADH can support the full range of engineering activities and analyses, from conceptual design through detailed analysis and validation.

A critical aspect of this comprehensive representation is the seamless sharing of data between different engineering disciplines and their respective tools. The ADH serves as a central repository for aircraft data, ensuring consistency and accuracy across various aspects of aircraft design and analysis. This centralized approach eliminates the redundant data entry and inconsistencies that characterize current, siloed data management approaches.

10.3 Work Breakdown Structure Integration

While Air Transport Association (ATA) chapters provide a standardized categorization for commercial aircraft systems, a Work Breakdown Structure (WBS) approach offers superior flexibility and granularity for organizing a comprehensive aircraft data repository. By breaking down the aircraft and its associated data into hierarchical levels representing systems, sub-systems, components, and even specific data types (e.g., maintenance records, engineering drawings, performance data), the WBS allows for more detailed and context-rich organization. This structure facilitates complex queries, cross-referencing between data types, and the integration of project-related information, such as maintenance schedules, cost tracking, and modification projects. Additionally, a WBS can adapt to the evolving needs of the repository, accommodating new data types or organizational structures beyond the rigid framework of ATA chapters, making it a more versatile and scalable solution.

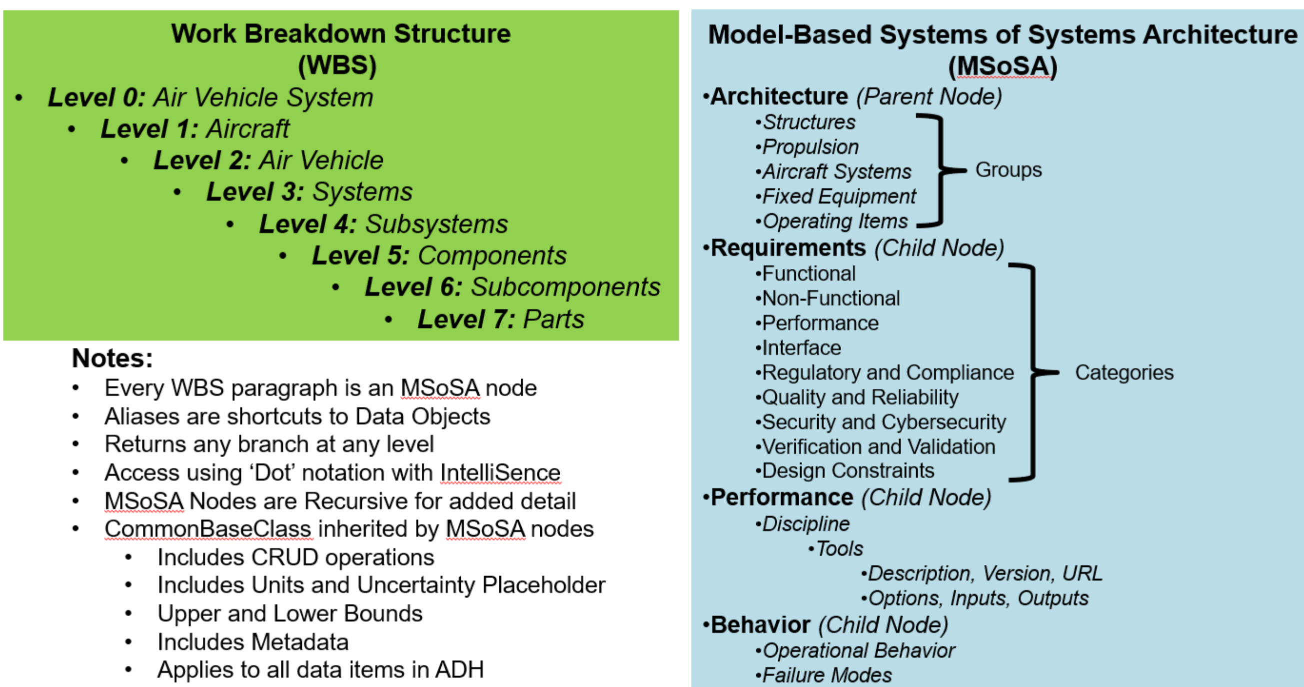

Figure 10- Illustration of the Work Breakdown Structure (WBS) and Model-Based Systems of Systems Architecture (MSoSA) for the Aircraft Data Hierarchy (ADH), detailing the hierarchical levels and organizational guidelines.

The Work Breakdown Structure (WBS), based on MIL-STD-881F with modifications to align with other relevant industry standards, provides the top-level organizational framework for the ADH. This structure divides the aircraft into logical components and systems, creating a comprehensive, organized representation of the entire air vehicle system, as shown in Figure 10. While acknowledging the historical similarities and potential applicability of ATA Chapters, we ultimately chose the MIL-STD-881F summarized in Figure 11.

This decision stemmed from the understanding that our military customer would likely demonstrate stronger buy-in and familiarity with a military standard. Consequently, leveraging a structure aligned with their established practices ensures smoother adoption and integration of the ADH, given the WBS's applicability to commercial applications as well.

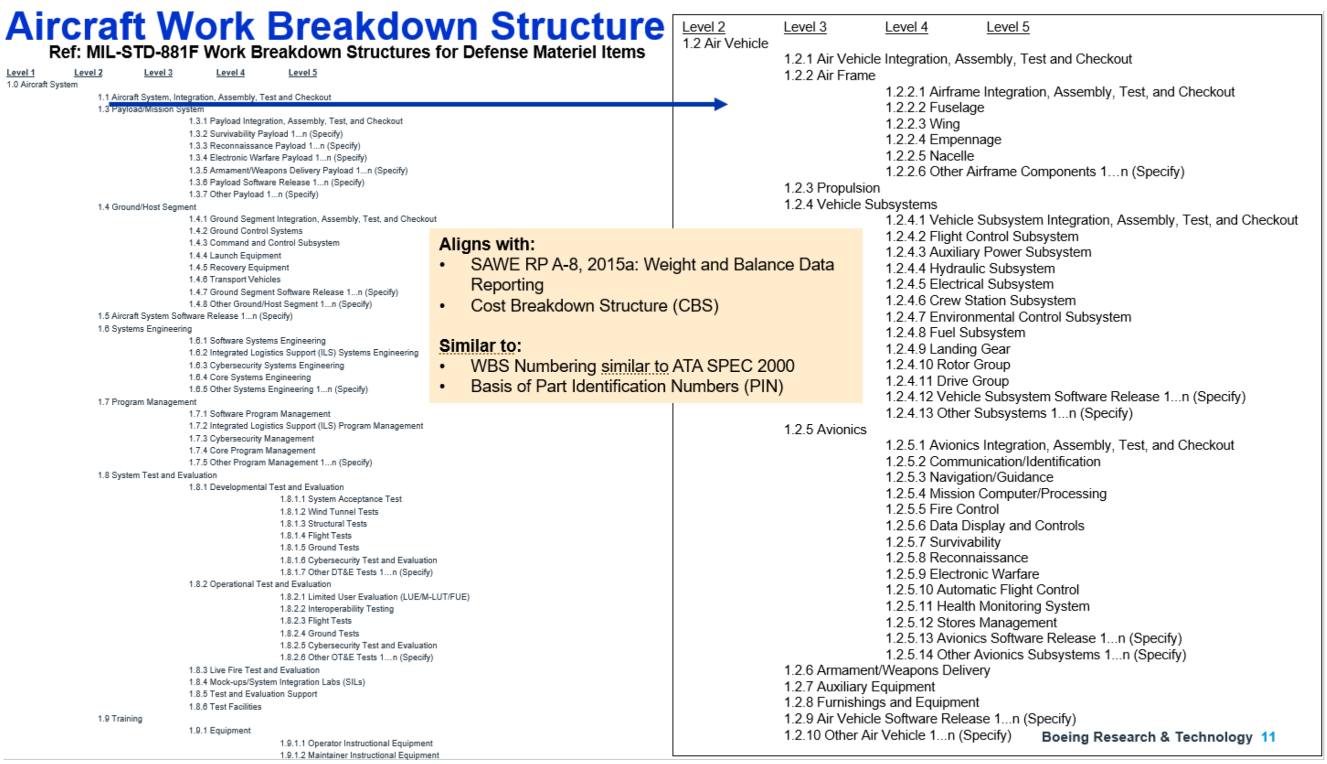

Figure 11- Detailed view of the Aircraft Work Breakdown Structure (WBS) based on MIL-STD-881F, illustrating the hierarchical levels and alignment with industry standards.

10.4 Hierarchical Organization through MSoSA

The ADH aligns with the Model-Based System-of-Systems Architecture (MSoSA) guidelines defined in Figure 12, which defines a hierarchical structure with the Architecture view as the parent and Requirements, Performance, and Behavior as child views. This parent-child relationship ensures that all analysis, requirements, and behavioral characteristics are derived from and traceable to the fundamental architectural definition.

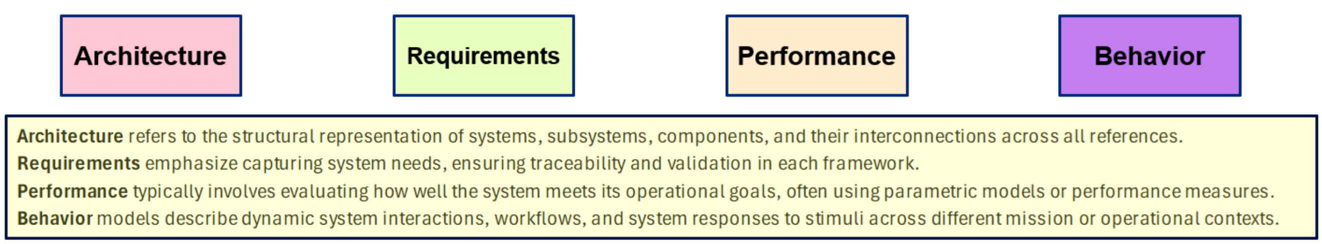

Figure 12- Overview of the four key views in the ADH framework: Architecture, Requirements, Performance, and Behavior, highlighting their roles and definitions.

The recursive MSoSA class structure is repeated for every level (paragraph) of the WBS structure, creating a consistent, hierarchical representation of the Air Vehicle System down to individual aircraft components and subsystems. This consistency simplifies navigation and understanding of the data hierarchy, enhancing usability for engineers and analysts.

The Architecture view defines the fundamental characteristics of each component and its relationship with the system and other components, including shape, position, and interfaces. This architectural foundation provides the basis for all subsequent analyses and evaluations, ensuring consistency and traceability throughout the design process.

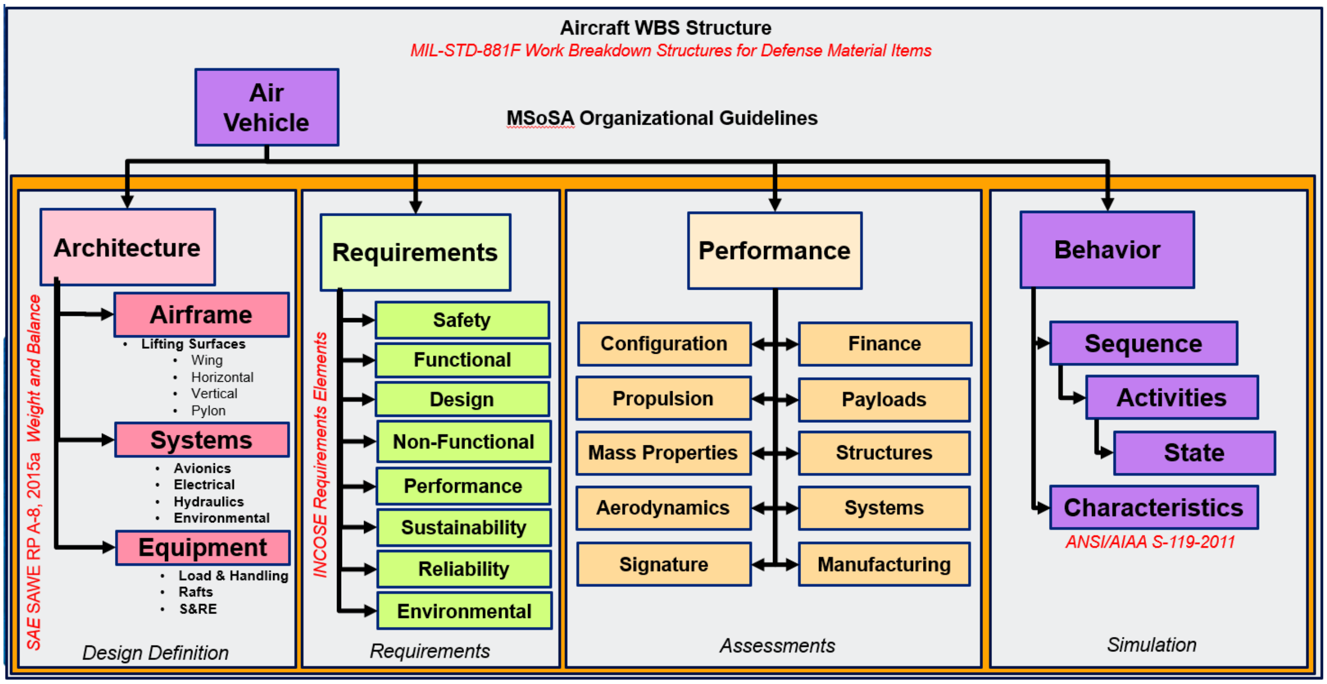

Figure 13- Diagram of the Aircraft WBS Structure aligned with MSoSA Organizational Guidelines, illustrating the hierarchical organization of Architecture, Requirements, Performance, and Behavior views.

The child views—Requirements, Performance, and Behavior—derive from and reference the parent Architecture view, creating a cohesive and integrated representation of the aircraft system. This hierarchical organization aligns with established systems engineering methodologies, providing a natural framework for implementing model-based systems engineering approaches as demonstrated in Figure 13.

10.5 Principle of Similar Treatment

A fundamental principle of the ADH is that similar components should be treated similarly in both their data representation and behavioral analysis. This principle recognizes that aircraft systems exhibit recurring architectural patterns, with natural groupings of components that share common characteristics and behaviors. For Model-Based Systems Engineers, this principle promotes conceptual clarity and consistency, reducing the cognitive overhead associated with understanding and managing diverse aircraft components.

The ADH identifies and leverages these natural groupings, such as axial components (nacelles, pods, fuselages) and aerodynamic surfaces (wings, tails, canards), to create a coherent, intuitive structure for aircraft data representation. This approach not only enhances conceptual clarity but also promotes programming efficiency and knowledge transfer, as similar components can share common data structures and behaviors.

10.6 Dictionary of Aliases

The lowest level 'leaves' of the ADH provide the structure of the individual data objects, where specific parameters and properties are defined. This level of detail ensures that the ADH can capture all relevant information for the comprehensive design and analysis of the air vehicle system, while maintaining alignment with established industry standards. A dictionary of shortcuts to the individual MSoSA data objects in the ADH, shown in Figure 14 are called aliases. These aliases are shorthand notations for specifying data paths, simplifying data access for code developers.

For Model-Based Systems Engineers at NASA, this philosophical foundation provides a robust, intuitive framework for implementing sophisticated design methodologies. The ADH's emphasis on natural patterns, comprehensive representation, hierarchical organization, and standards alignment creates a data management approach that enhances rather than impedes the engineering process, enabling more effective and efficient aircraft design and analysis.

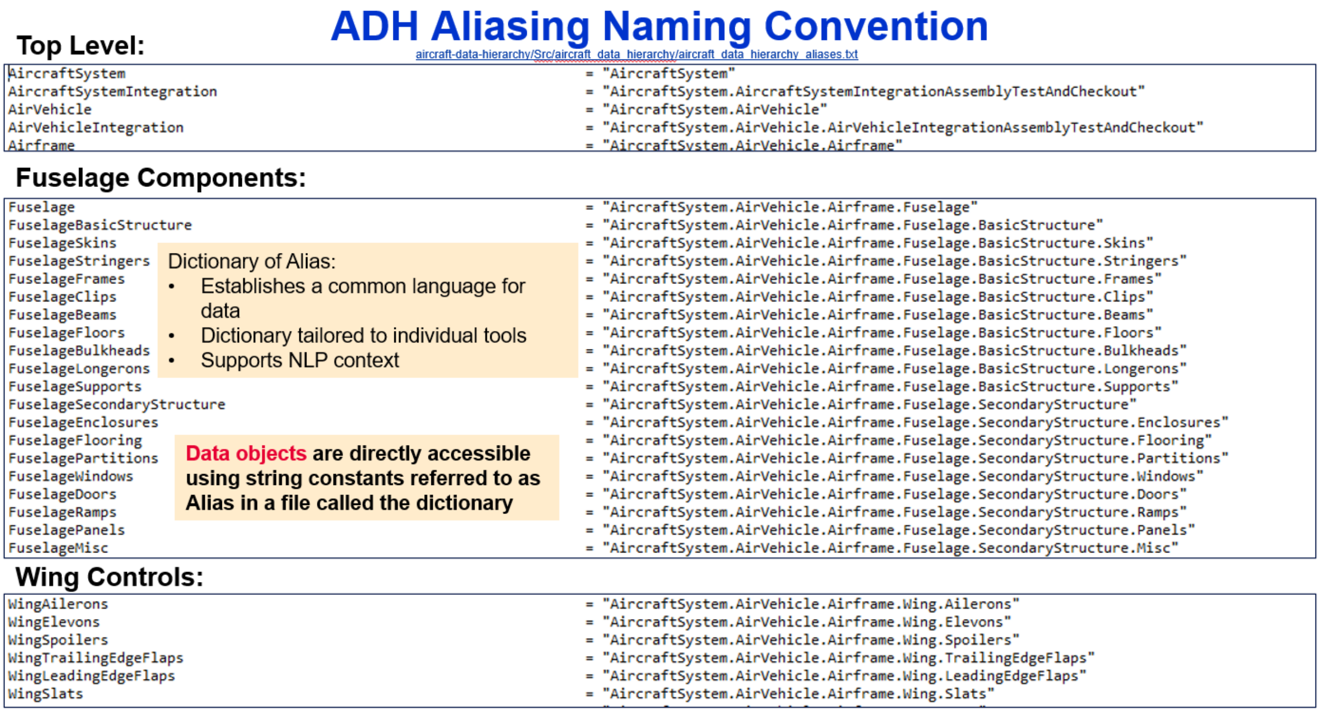

Figure 14- ADH Aliasing Naming Convention, illustrating the hierarchical structure and dictionary of aliases for accessing data objects within the Aircraft Data Hierarchy.

11.0 Benefits of the ADH

The Aircraft Data Hierarchy delivers substantial benefits across every aspect of the aerospace design process, addressing critical challenges faced by Model-Based Systems Engineers in industry and providing capabilities that significantly enhance design efficiency, accuracy, and collaboration.

11.1 Standardization and Consistency

ADH provides consistent parameterization aligned with established aerospace standards, ensuring uniformity across the data hierarchy. This standardization creates a common language for aircraft design and analysis, reducing ambiguities and miscommunications that often plague cross-disciplinary collaborations. Data is organized in a logical hierarchy, aligned with familiar standards such as MIL-STD-881F and SAWE RP A-8, facilitating intuitive navigation and access for engineers from different disciplines.

For Model-Based Systems Engineers, this standardization is transformative. It eliminates the need to translate between different data representations and terminologies, streamlining the integration of diverse analyses and ensuring consistent interpretation of requirements and results across the entire design process. This consistency significantly reduces the time and effort required to coordinate across disciplinary boundaries.

11.2 Flexibility and Adaptability

The framework is inherently modifiable to meet various design needs, ensuring flexibility across different scenarios and applications. This adaptability is essential in aerospace design, where diverse aircraft configurations, from conventional fixed-wing designs to novel urban air mobility concepts, require different data representations and analyses. The ADH accommodates this diversity while maintaining a consistent overall structure, balancing standardization with the flexibility necessary for innovation.

As aircraft designs grow in complexity and incorporate new technologies, the ADH scales effectively to accommodate this evolution. This scalability ensures that the data hierarchy supports varining fidelity throughout the entire design lifecycle, from initial concept development through detailed analysis and validation. For Model-Based Systems Engineers managing complex, integrated systems, this scalability is essential for maintaining design coherence as the level of detail increases.

11.3 Enhanced Performance and Reliability

ADH provides reliable performance, especially in the early design phases where data accuracy and consistency are critical for establishing a solid foundation for subsequent development. The schema of the ADH defined as Pydantic classes include comprehensive validation capabilities that ensure data integrity and identify potential issues early in the design process, when they are less costly to address.

The detailed requirements captured in the ADH ensure thorough documentation of the aircraft design, creating a comprehensive and reliable resource for design reviews, analysis, and decision-making. This documentation supports rigorous systems engineering processes, providing the traceability and evidence necessary for verification and validation activities.

11.4 Seamless Integration and Collaboration

The ADH's interoperability with existing analysis tools facilitates seamless integration into current workflows and processes, minimizing the disruption associated with adopting a new data standard. This integration extends across the entire tool ecosystem, from conceptual design tools through detailed analysis and simulation environments, creating a continuous digital thread throughout the lifecycle of the aircraft.

By providing a centralized, shared data repository, the ADH promotes collaboration among project teams and stakeholders. This shared resource enhances communication and ensures that all team members are working from the same, consistent data, eliminating discrepancies that can lead to design conflicts and integration issues. For Model-Based Systems Engineers coordinating across multiple disciplines, this collaborative capability is transformative, enabling more efficient, effective teamwork.

11.5 Operational Improvements

The ADH is architected to eliminate duplication and reduces error across design and assessment workflows, promoting data integrity and reliability. The single-source-of-truth approach ensures that changes are consistently propagated throughout the design, reducing the risk of inconsistencies and outdated information. This consistency is particularly valuable in complex design environments where changes in one area can have cascading effects across multiple systems and analyses.

Maintaining the integrity of the ADH data repository requires its distinct separation from the integration processes and tools used to access aircraft behavioral characteristics. This separation enables reuse of the ADH for the automation of simple trade studies and more complex MDAO studies.

Shared data enhances cross-disciplinary communication throughout the design process, fostering collaboration and knowledge exchange. Engineers from different disciplines can access and understand data from other domains, creating opportunities for innovation and optimization that might be missed in more siloed approaches. This enhanced communication is especially valuable for systems engineers who must coordinate across disciplinary boundaries.

11.6 Enhanced Traceability and Automation

Logging the transactions of data flowing in and out of the ADH provides better change tracking and requirements fulfillment throughout the design cycle, improving project management and accountability. A Digital Thread can be implemented by logging each design decision and its impact on requirements can be documented and traced, creating a comprehensive history of the design evolution. This traceability supports rigorous systems engineering processes and facilitates regulatory compliance and certification activities.

Automated execution, updates, and validation streamline processes and improve efficiency, reducing manual effort and potential errors. These automation capabilities transform time-consuming, error-prone manual processes into efficient, reliable automated workflows, allowing engineers to focus on high-value design activities rather than administrative data management tasks.

11.7 System-Level Insights and Digital Continuity

A modern unified system view and enhanced data analytics support more informed decision-making, providing holistic insights that might be missed in more manual approaches. This comprehensive perspective is essential for systems engineers who must balance competing requirements and optimize key system level performance behavior, not just individual component characteristics.

The ADH ensures lifecycle data integration, supporting a continuous digital thread that maintains data consistency and integrity throughout the aircraft lifecycle. This continuity extends from initial concept development through detailed design, manufacturing, operations, and provides a foundation for Digital System Models (DSM) and lifecycle management. Real-time data alignment enables accurate monitoring and control, supporting advanced applications like Digital Twins and facilitating dynamic, data-driven decision-making. For Model-Based Systems Engineers at NASA, these benefits represent a significant advancement in the tools and methodologies available for aircraft design and analysis. The ADH transforms data management from a necessary administrative burden into a strategic advantage, enabling more efficient, effective, and innovative approaches to aerospace engineering.

12.0 Implementation

12.1 Technical Detail

The Aircraft Data Hierarchy employs a sophisticated technical architecture designed to meet the rigorous demands of aerospace data management while ensuring compatibility with existing systems and processes. This implementation leverages best-in-class technologies and methodologies to create a robust, extensible framework for aircraft data representation and exchange. This architecture can be seen partially in Figure 15 which shows a part of the ADH visualized through a web editor.

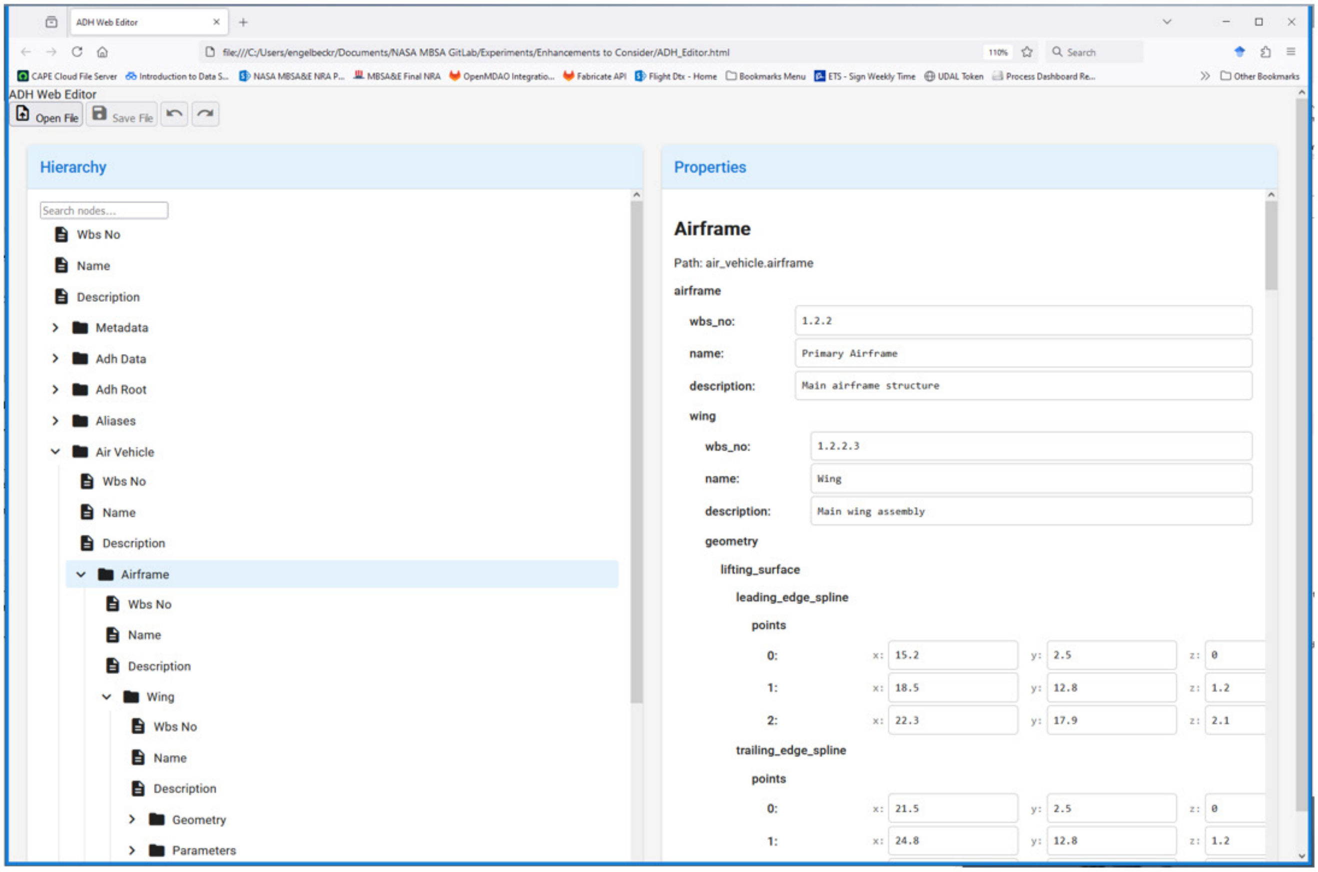

Figure 15- Screenshot of the ADH Web Editor, displaying the hierarchical structure and properties of the Airframe data object within the Aircraft Data Hierarchy.

The ADH has been released to the open-source community to facilitate further development (https://github.com/Boeing/aircraft-data-hierarchy) (see Figure 16). Collaboration between the aerospace industry and NASA in a GitHub environment offers the best opportunity to create an innovative, transparent, and enduring aircraft data exchange standard. Adhering to best software engineering practices, such as PEP8 guidelines, unit tests, and robust automated documentation, will maximize the quality of the ADH implementation. Additionally, ensuring that the naming conventions for paths and variables in the ADH are descriptive will provide the necessary context for Natural Language Processing (NLP) and Generative Artificial Intelligence (GenAI) applications in the near future.

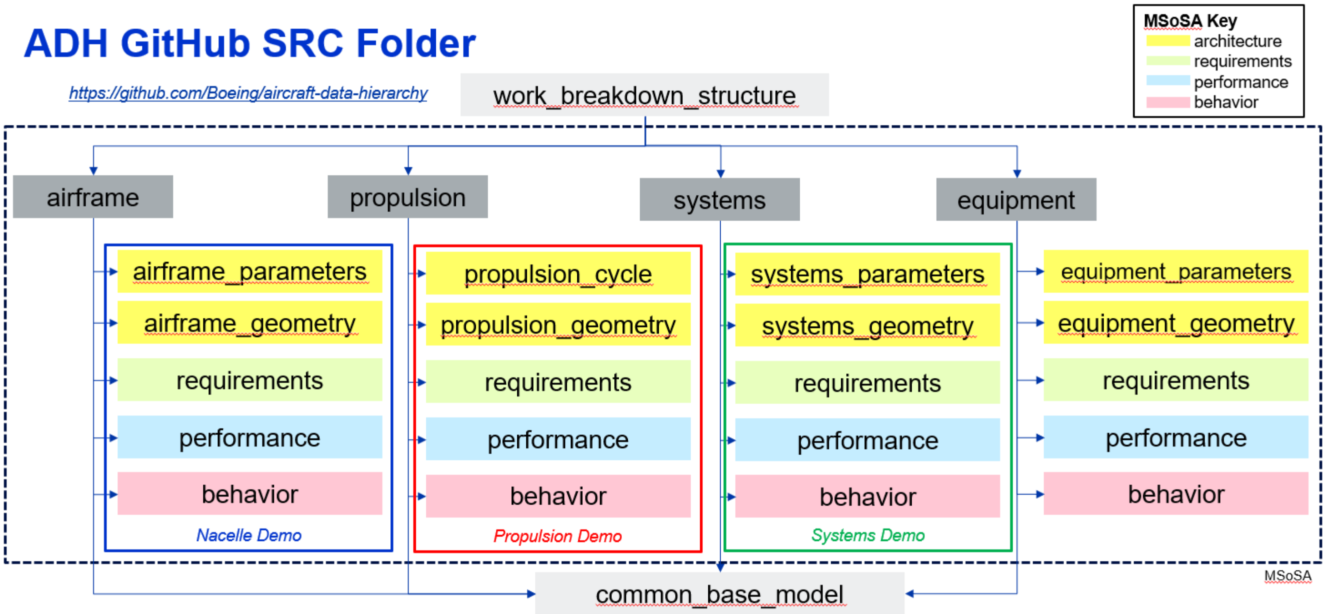

Figure 16- ADH GitHub SRC Folder structure, illustrating the organization of data objects by airframe, propulsion, systems, and equipment, aligned with the MSoSA framework.

12.2 NASA MBSA&E Computational Framework and MBSE Integration

MagicDraw/Cameo is the MBSE tool of choice for Model-Based Systems Engineers. OpenMDAO, NASA's open-source framework for multidisciplinary design, analysis, and optimization, serves as the computational backbone for NASA's MBSA&E framework. It offers a flexible and powerful platform for integrating complex systems with numerous interconnected components.

The data hierarchy in the ADH persists the MBSE data metamodel. Integrating MagicDraw with the ADH to create the SysML data metamodel representation bridges the gap between it and NASA's OpenMDAO framework. Integrating the Standard Evaluator with the ADH completes the end-to-end integration between MBSE and OpenMDAO, defining NASA's MBSA&E framework shown earlier in Figure 3. This ensures seamless compatibility with existing NASA workflows and tools, minimizing the disruption associated with adopting a new data standard.

This integration facilitates the implementation of comprehensive model-based systems engineering methodologies, ensuring consistent data representation across different analysis domains that are re-configurable without requiring re-integration.

12.3 Programming Language and Data Validation

Python serves as the primary programming language for implementing the ADH, chosen for its versatility, extensive library ecosystem, and widespread adoption in scientific and engineering communities. This choice aligns with NASA's existing software infrastructure and provides access to a rich ecosystem of tools and libraries for data manipulation, analysis, and visualization.

A key innovation in the ADH implementation is the use of Pydantic v2 for schema definition and validation. Another example is in Figure 17. Pydantic provides robust, runtime data validation with automatic type checking, ensuring that data within the ADH conforms to the defined schema. This validation is critical for maintaining data integrity and consistency, particularly in complex, collaborative design environments where data may be generated and modified by multiple tools and users.

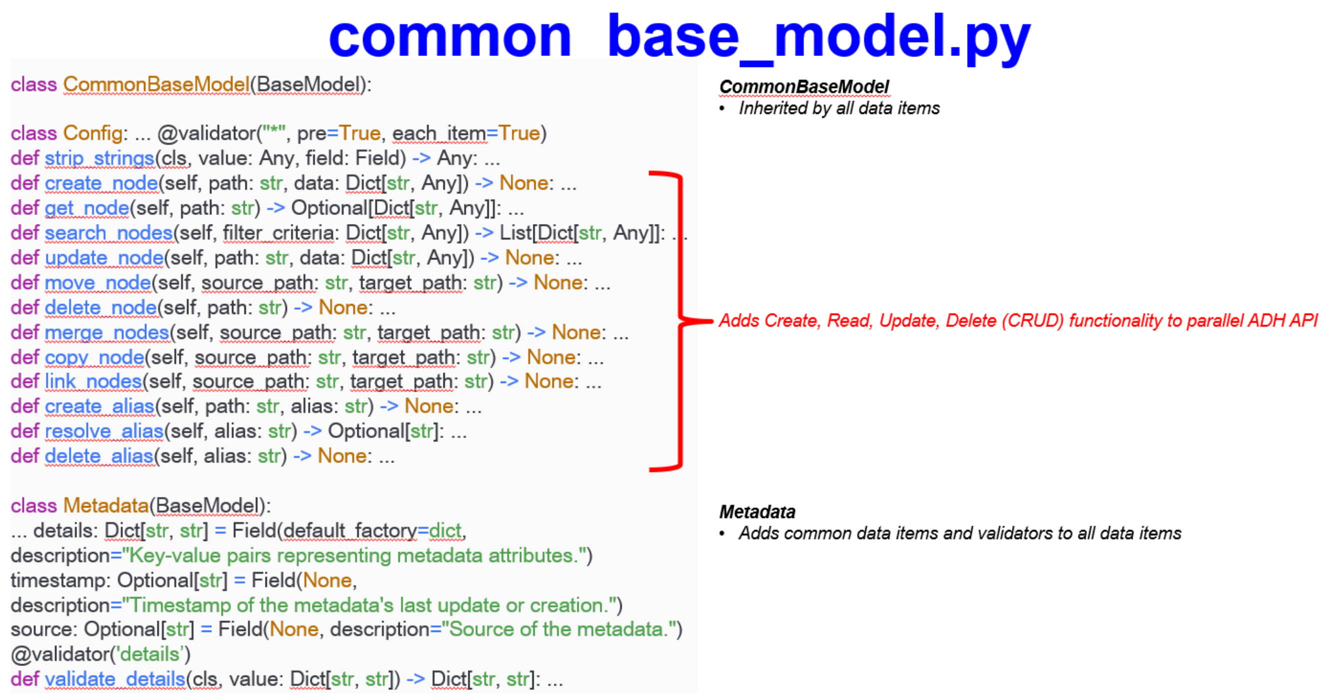

Figure 17- Overview of the common_base_model.py file, illustrating the CommonBaseModel class with CRUD functionality and metadata validation for the ADH framework

The choice of Pydantic also positions the ADH for future integration with advanced analytics and artificial intelligence applications. Pydantic has emerged as a common data management choice for many machine learning and Generative AI frameworks, including FastAPI, HuggingFace Transformers, Langchain, and BentoML. This alignment creates opportunities for leveraging these advanced technologies as they mature and gain acceptance in the aerospace design process.

12.4 Data Formats and Persistence

The ADH supports multiple data formats for persistence and exchange, including XML, JSON, and YAML. This flexibility allows engineers to choose the format that best suits their specific needs and existing toolchains, reducing the barriers to ADH adoption. Each format offers specific advantages:

- JSON, our choice for the persistence format of the ADH, offers excellent compatibility with web technologies and programming languages, with efficient parsing and compact representation, making it ideal for real-time data exchange and integration with web-based tools and networks.
- XML, the persistance format for CPACS and ADML, provides a mature, widely-supported format with robust schema validation capabilities, making it suitable for formal data exchange and archival purposes.
- YAML provides a human-readable format with support for complex data structures, making it well-suited for configuration files and documentation.

This multi-format capability ensures that the ADH can integrate seamlessly with a diverse range of existing tools and workflows, providing flexibility for different use cases and preferences.

12.5 Development Tools and Quality Assurance

The development of the ADH employs a comprehensive set of GitHub tools to ensure code quality, version control, and collaborative development. GitHub provides a unified platform for version control, issue tracking, and continuous integration/continuous deployment (CI/CD), facilitating efficient collaboration among distributed development teams.

Although we did not use code quality tools on the prototype ADH project, we recommend incorporating SonarQube and Coverity for the production ADH codebase. These tools provide automated scanning and analysis, identifying potential issues related to security, reliability, and maintainability. By using these tools, we can ensure that the production ADH maintains high standards of code quality and adheres to established best practices for software development.

Auto-generation tools for creating schema-compliant data access classes enhance development efficiency and ensure consistency in the implementation of the data hierarchy. These tools reduce the manual effort required to implement and maintain the ADH codebase, allowing developers to focus on higher-level architecture and functionality.

12.6 Organizational Structure

The ADH is organized according to MIL-STD-881F, with modifications to align with SAWE RP A-8. This organizational structure provides an overarching framework for representing and managing aircraft data, based on established industry standards. The Work Breakdown Structure (WBS) provides a logical, hierarchical organization for the data, facilitating efficient navigation and access commonly followed by Mass Properties and Cost Accounting tools.

Each line item in the WBS is implemented as a recursive Model-Based Systems Architecture (MSoSA) object, representing air vehicle components, systems, and equipment. This recursive structure enables consistent representation of complex, nested systems while maintaining clarity and organization. For Model-Based Systems Engineers, this structure provides a familiar, intuitive framework for organizing and accessing aircraft data.

13. Current Progress and Achievements

Significant progress has been made in the development of the Aircraft Data Hierarchy, establishing a solid foundation for this transformative framework. The team has completed several key milestones that demonstrate the viability and value of the ADH approach.

13.1 Foundational Development

The initial phase focused on establishing the essential infrastructure for ADH development, including the development environment and coding standards. These foundational elements ensure consistency and quality in the implementation of the ADH, creating a solid foundation for continued development.

A critical early milestone was a comprehensive kickoff meeting with NASA stakeholders, which aligned project goals and requirements and created a shared understanding of the project's objectives and approach. This alignment ensures that the ADH development is focused on addressing NASA's specific needs and priorities.

13.2 Requirements Gathering and Stakeholder Engagement

Comprehensive interviews and discussions with some industry stakeholders (NASA, Boeing, GE, Collins, UofM) have been conducted to finalize the data requirements for the ADH. These discussions provided valuable insights into the needs and expectations of potential users, informing the design and functionality of the data hierarchy. This stakeholder-driven approach ensures that the ADH addresses real-world challenges and provides tangible benefits for its users.

Based on these requirements, detailed use cases (demonstrations) were developed to guide the development process, ensuring that the ADH addresses specific, practical scenarios encountered in aerospace design. These use cases provide concrete examples of how the ADH will be used in practice, informing design decisions and implementation priorities.

13.3 Architectural Decisions and Schema Development

High-level architectural decisions have been made that will guide continued development of the data hierarchy. These include the selection of the MIL-STD-881F Work Breakdown Structure for the high-level structure of the ADH and the adoption of the MSoSA guidelines for organizing and representing aircraft data.

A significant achievement has been the alignment of many behavioral data nodes in the ADH with ANSI/AIAA-S-119-2011 Simulation Model Exchange Standard. This alignment ensures compatibility with established industry practices and facilitates interoperability with existing systems and tools.
The team has delivered a draft ADH Pydantic schema, providing a tangible implementation of the data hierarchy concepts. This schema defines the structure and relationships of the ADH data objects, creating a foundation for validation and enforcement of data integrity.

13.4 Prototyping and Integration

Prototyping efforts have encompassed several key areas, including schema development, API implementation, and integration testing. The prototype schema has been iteratively refined based on feedback from stakeholders and integration tests, ensuring that it effectively meets the needs of its users.

The development of a prototype Python package for accessing and managing data within the Aircraft Data Hierarchy provides essential functionality for interacting with the data hierarchy. Basic helper functions have been implemented and documented for initial testing, demonstrating the practical utility of the ADH.

Initial integration tests with the MBSA&E model demonstrated in CLIN-003 have validated the seamless interaction of the ADH is possible with Magicdraw as a critical MBSE tool. These tests have identified areas for improvement and additional functionality required for full integration, informing ongoing development efforts.

13.5 Demonstration of Capabilities

To validate the functionality and benefits of the ADH, several demonstrations have been conducted, showcasing its capabilities in real-world scenarios. These demonstrations provide tangible evidence of the ADH's value and potential impact on aerospace design processes.

A walkthrough of the ADH structure and functionality provided stakeholders with a detailed understanding of the data hierarchy, its organization, and its capabilities. This demonstration facilitated feedback and refinement of the framework, ensuring that it meets user expectations and requirements.

A specific demonstration of the ADH using a nacelle object highlighted the structure and functionality of the data hierarchy for a concrete aircraft component. This demonstration showcased how the ADH can effectively represent and manage data for specific aircraft elements, providing a practical example of its application and the use of Intellisence.

Exploration of bidirectional data exchange between Magic Systems Cameo MBSE models and the ADH highlighted the integration and interoperability capabilities of the framework. This demonstration showed how the ADH can serve as a bridge between MBSE and MBSA sides of the MBSA&E framework facilitating seamless data exchange and maintaining consistency across diverse tools and processes.

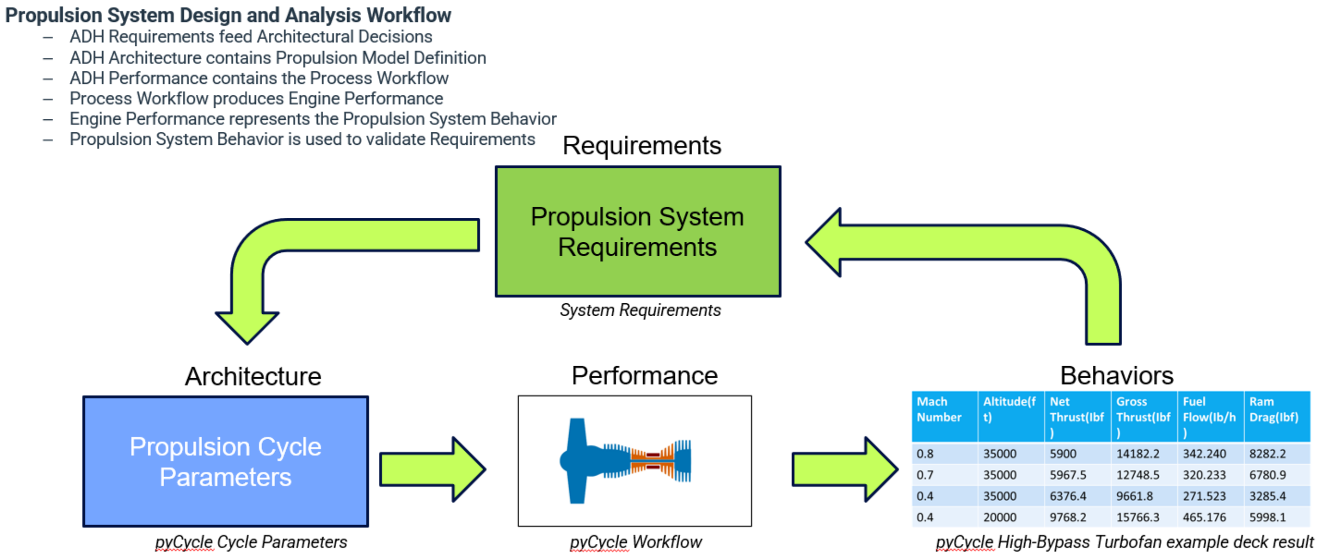

Figure 18- Diagram of the Propulsion System Design and Analysis Workflow within the ADH framework, illustrating the interaction between Requirements, Architecture, Performance, and Behaviors.

The propulsion demonstration development began by defining the ADH architecture, behavior, and performance branches with the parameters required for the pyCycle zero-dimensional engine performance analysis with our GE Aerospace partner.

- **Architecture Branch:** This branch includes Pydantic classes for each individual engine component, such as compressors, ducts, and turbines, along with all the design parameters related to each respective component. The parent class (architecture) contains general information about the engine cycle itself.
- **Behavior Branch:** This branch mirrors the architecture branch but contains off-design performance at user-specified design conditions. It includes classes for each engine component and cycle.
- **Performance Branch:** This branch contains a Pydantic tool class that enables the user to specify solver setting inputs, outputs and options controlling the pyCycle propulsion analysis.

The next step in the propulsion analysis demonstration involves using the propulsion analysis builder tool referenced in the performance branch. This builder tool accepts an ADH instance as input and outputs an OpenMDAO pyCycle model ready for analysis. The builder tool features a parent class with methods to gather all propulsion-related parameters from the ADH and a child class that generates the OpenMDAO pyCycle model. This parent-child architecture allows for the future addition of support for other propulsion analysis tools with relative ease.

After generating the OpenMDAO pyCycle model, users must create their own script to set initial guesses for the solver and then run the analysis with their chosen order of flight conditions. The ADH package contains helper functions that write the resulting engine deck back to the behavior branch of the ADH instance and/or create a JSON-formatted data file. These helper functions can write multiple engine decks to the ADH if the user wishes to consider multi-engine analyses.

The integration of the ADH with a propulsion system design and analysis workflow demonstrated its utility in a complex, multidisciplinary scenario. This workflow, shown in Figure 18, involved establishing requirements, developing the architecture, defining performance metrics, executing the process workflow, generating engine performance data, and validating the requirements—all facilitated by the ADH's comprehensive data representation.

After executing pyCycle, a helper function can be called to import the propulsion table data into the pre-existing ADH while maintaining the proper Pydantic class structure. This ensures that the data remains consistent and properly validated within the ADH environment.

Additionally, another helper function has been implemented to export an engine deck JSON file from the ADH, formatted for compatibility with Aviary's software. The DaveML format has been integrated with the Pydantic class structure to represent the DaveML standard. The engine deck is generated using the DaveML Pydantic classes in the ADH library, specifically following the ANSI/AIAA-S-119-2011 specification. This standard ensures data clarity, consistency, and interoperability between various aerospace modeling and simulation tools. By leveraging DaveML's ungridded data table standard, the function constructs an engine table representation that is both accurate and easy to interpret.

To facilitate seamless integration with Aviary, a new function has been introduced into Aviary's engine table reading class. This enhancement allows Aviary to parse and read engine deck JSON files that use the DaveML format directly. The implementation has undergone validation using Aviary's Level 2 interface, successfully running test cases and confirming compatibility. Users can easily run Aviary with the generated engine deck by specifying the file path within the input .csv file used for the run.

This streamlined workflow, from pyCycle to ADH and finally to Aviary, showcases the robustness and flexibility of the ADH's data management capabilities. The ability to export ADH engine table data into a standardized DaveML format and import it directly into Aviary highlights the potential for further integration with external tools, facilitating enhanced collaboration and accelerated aerospace design and analysis processes.

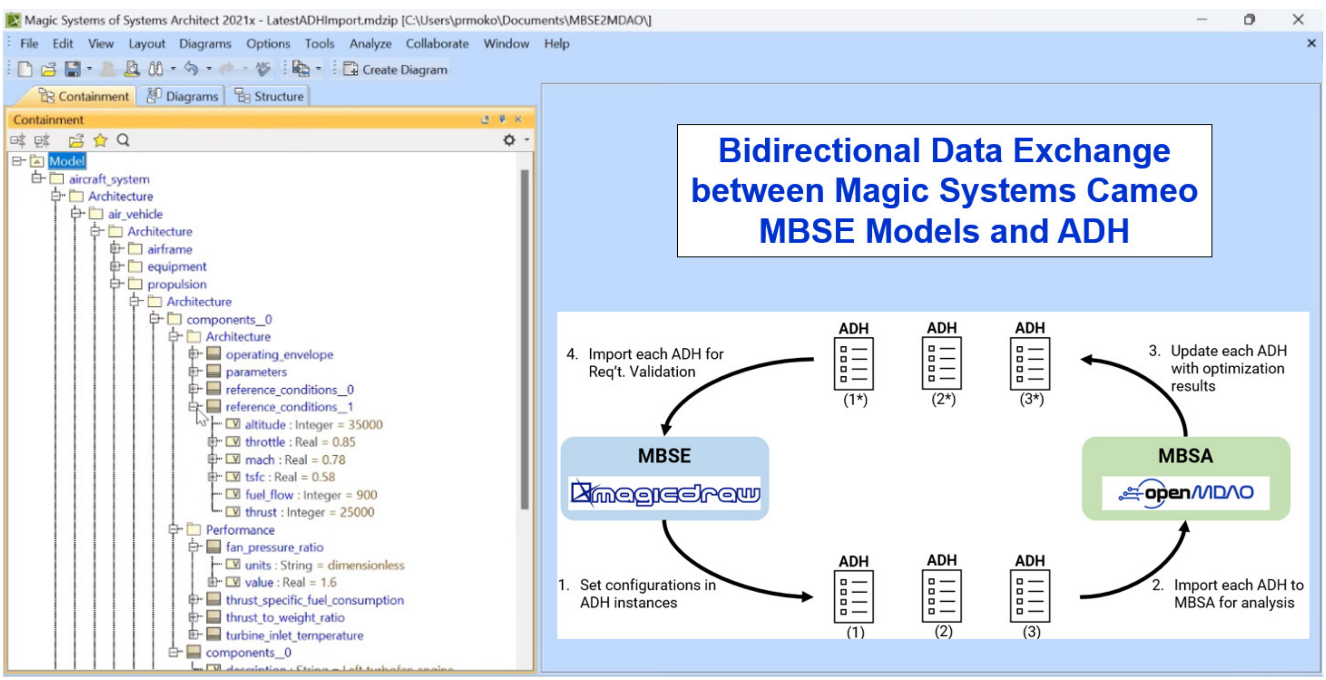

Figure 19- Bidirectional Data Exchange between Magic Systems Cameo MBSE Models and ADH, illustrating the workflow for setting configurations, importing data, updating optimization results, and validating requirements.

After running an analysis using the ADH data, it is imperative to verify that the system architecture satisfies the design requirements. Magicdraw can be used to accomplish this. To connect the analysis output (or any ADH) to Magic Systems Cameo, an interface was written to convert a JSON string into a system model. This is done by serializing the modified ADH to a JSON file and then calling one of the interface functions within Magic Systems Cameo. Conversely, the de-serializing process can be performed to convert the system model to an ADH. Thus, there is a bidirectional relationship between the MBSE and MBSA tools that interact with the ADH. Figure 19 illustrates a cyclic process between the ADH inputs/outputs changes and the actions performed within Magic Systems Cameo.

In the workflow, each ADH represents a different or updated aircraft instance to be analyzed by an external tool. Each of these may be exported separately from Magic Systems Cameo and run by an MBSA tool. After running the analysis in MBSA, the results can be imported back into Magic Systems Cameo for requirement verification. This process may be repeated for any set of systems, subsystems, or components in the system architecture. Additional information on the bidirectional interaction between the ADH and Magic Systems Cameo may be found in the CLIN-003 Report, which details the interface functions developed to support these capabilities. The integration of the ADH with a propulsion system design and analysis workflow demonstrated its utility in a complex, multidisciplinary scenario. This workflow involved establishing requirements, developing the architecture, defining performance metrics, executing the process workflow, generating engine performance data, and validating the requirements—all facilitated by the ADH's comprehensive data representation.

These achievements represent significant progress in the development of the Aircraft Data Hierarchy, establishing a solid foundation for continued advancement and refinement. For Model-Based Systems Engineers at NASA, these accomplishments demonstrate the practical viability of the ADH approach and its potential to transform aerospace data management and exchange.

14. Industry Involvement and Collaboration

The success of the Aircraft Data Hierarchy depends not only on its technical merits but also on its adoption and support by the broader aerospace community. Recognizing this, the ADH development has actively engaged with key stakeholders across the industry, creating a collaborative ecosystem that ensures the framework meets diverse needs and leverages industry expertise.

14.1 Key Collaborators

The development of the prototype ADH has involved collaboration with several key aerospace organizations, including NASA, Boeing, GE Aerospace, Collins Aerospace, and the University of Michigan. This diverse group of stakeholders provides a comprehensive perspective on the requirements and challenges of aerospace data management across different sectors and applications.

Boeing envisions expanding NASA’s current sponsorship of the ADH project into a future leadership role that will drive the evolution of aerospace data management. Building on its foundational expertise in aerospace systems and multidisciplinary design optimization, NASA would lead the ADH initiative by leveraging the American Institute of Aeronautics and Astronautics (AIAA) to engage more deeply with the aerospace community. This expanded role will involve orchestrating a collaborative ecosystem that includes industry leaders, academic researchers, and government agencies. By fostering open dialogue and the exchange of innovative ideas, NASA will ensure that the ADH evolves to meet the dynamic needs of the aerospace sector. This vision includes the development of new standards, the integration of emerging technologies, and the promotion of best practices across the industry. Ultimately, NASA's leadership will position the ADH as a cornerstone of aerospace data management, driving advancements that enhance efficiency, interoperability, and innovation across the entire aerospace community.

Engaging with additional universities would bring significant benefits to the development of the ADH. Universities are hubs of cutting-edge research and innovation, and their involvement would introduce fresh perspectives and novel approaches to aerospace data management. Academic institutions can contribute advanced theoretical knowledge, experimental research, and innovative methodologies that can enhance the ADH's capabilities. Furthermore, collaboration with universities would facilitate the training and development of the next generation of aerospace engineers, ensuring that they are well-versed in the latest data management standards and practices. This partnership would also foster a continuous exchange of ideas between academia and industry, driving ongoing improvements and ensuring that the ADH remains at the forefront of technological advancements.

The future involvement of defense organizations like AFRL, NAVAIR, and DARPA would brings valuable perspectives on military aircraft requirements and their use cases. These organizations have distinct needs and challenges, including specialized aircraft types, unique mission profiles, and specific regulatory and security considerations. Their involvement ensures that the ADH can accommodate these specialized requirements while maintaining a consistent, standardized approach.

Aircraft manufacturers and their suppliers play a pivotal role in the aerospace industry's digital transformation, bringing invaluable insights from aircraft development and production. Their involvement is essential for ensuring that the ADH addresses real-world challenges in aerospace design and seamlessly integrates with advanced digital processes and tools. Engaging prime aircraft manufacturers is critical because they possess deep expertise in the complexities of aircraft design, manufacturing, and certification; all of which are undergoing significant digital evolution.

By participating in the ADH project, prime manufacturers can help shape a cutting-edge data management framework that aligns with their digital transformation initiatives. This collaboration can lead to more efficient and agile design processes, reduced development costs, and enhanced product quality through the use of digital twins, predictive analytics, and other advanced technologies. A standardized data hierarchy will also facilitate better digital collaboration with suppliers, partners, and regulatory bodies, streamlining communication and minimizing the risk of errors.

Ultimately, the adoption of the ADH will empower prime aircraft manufacturers to fully leverage the benefits of digital transformation, driving innovation and efficiency in design and production. This alignment with digital transformation initiatives will position them as leaders in the aerospace industry, capable of delivering next-generation aircraft that meet the highest standards of performance, safety, quality and sustainability.

Open-sourcing the ADH project would bring numerous advantages, significantly enhancing its development and adoption. By making the ADH openly accessible, a broader community of developers, researchers, and industry professionals can contribute to its evolution, ensuring a more robust and versatile framework. Open-source collaboration fosters transparency, innovation, and rapid problem-solving, as diverse contributors bring unique insights and expertise. This collective effort can lead to the identification and resolution of issues more quickly, the addition of new features, and the continuous improvement of the ADH. Additionally, open-sourcing the project would encourage widespread adoption across the aerospace industry, academia, and other sectors, promoting standardization and interoperability. Ultimately, this collaborative approach would accelerate the advancement of aerospace data management, driving the ADH towards becoming a comprehensive and universally accepted standard.

14.2 Comprehensive Aircraft Coverage

The ADH is designed to cover all aircraft types and categories, providing a comprehensive solution for aircraft data management. This inclusive approach ensures that the framework can accommodate diverse aircraft configurations, from conventional fixed-wing designs to rotorcraft, unmanned aerial vehicles, and novel urban air mobility concepts.

Supporting this comprehensive coverage will be essential for creating a truly universal standard for aerospace data exchange. By accommodating the full spectrum of aircraft types, the ADH can provide a consistent framework that can be applied across different projects and programs, facilitating data exchange and collaboration even across diverse aircraft categories.

14.3 Open Source Development Model

The collaborative nature of the ADH development extends to its deployment through an open-source model. The team plans to publish the ADH on a public GitHub repository, inviting industry participation in detailing out the "leaves" of the ADH to address the specific needs of various tools and applications.

This open-source approach leverages the collective expertise and resources of the aerospace community, creating a more robust, comprehensive framework than would be possible through isolated development. By inviting industry participation, the ADH can evolve to address emerging needs and incorporate diverse perspectives and requirements.

14.4 Benefits of Collaborative Development

For Model-Based Systems Engineers at NASA, this collaborative development model offers several significant benefits. First, it ensures that the ADH addresses the full spectrum of aerospace design challenges, not just those encountered within NASA. This comprehensive perspective creates a more robust, versatile framework that can accommodate diverse use cases and requirements.

Second, industry involvement promotes adoption and integration of the ADH across the aerospace community. As key stakeholders contribute to and invest in the framework, they develop a sense of ownership and commitment that encourages implementation and use. This broad adoption creates a network effect, where the value of the ADH increases as more organizations and tools integrate with it.

Finally, collaborative development creates opportunities for knowledge exchange and innovation. As diverse organizations contribute their expertise and perspectives, the ADH benefits from a broader range of insights and approaches, leading to more creative, effective solutions to common challenges.

This collaborative approach positions the ADH not just as a NASA standard but as an industry-wide framework for aerospace data exchange. By engaging with key stakeholders across the aerospace community, the ADH development ensures that this framework will have the broad support and adoption necessary for transformative impact on aerospace design and analysis processes.

15. Next Steps and Future Vision

With the major overarching structure of the Aircraft Data Hierarchy now in place, the project is poised to enter its next phase of development, focusing on enhancing the detail, functionality, and impact of this transformative framework. For Model-Based Systems Engineers at NASA, this roadmap outlines the path toward a comprehensive solution that will address current challenges and establish a foundation for future innovation.

15.1 Detailing the Data Objects

A critical near-term priority is to work on the details of the individual data objects that form the "leaves" of the data hierarchy. With the high-level structure established, attention now turns to defining the specific parameters, properties, and relationships that characterize different aircraft components and systems. This detailed definition will ensure that the ADH provides comprehensive coverage of all relevant aspects of aircraft design and performance.

Three potential approaches have been identified for defining these data objects, each with different implications for timeframe, cost, and stakeholder involvement:

1. Tool-Driven Definition: Allow the first tools integrated with the ADH to determine the structure of the data objects. This approach offers the fastest path to implementation but may result in a less comprehensive or balanced representation, and less buy-in across industry.
2. ADML-Based Definition: Start with the well-defined ADML parameterization as a foundation. This approach leverages existing work and provides a solid starting point that can be adapted and extended to meet industries specific requirements.

3. CPACS-Based Definition: Start with the well-established CPACS parameterization as a foundation. This approach leverages existing work and provides a solid starting point that can be adapted and extended to meet industries specific requirements.
4. Industry Consensus Definition: Involve industry stakeholders in forming a consensus definition of the individual data objects. While this approach requires the most time and coordination, it offers the potential for a truly comprehensive, widely-accepted standard.

The selection among these approaches will depend on NASA's priorities regarding timeframe, comprehensiveness, and industry adoption, and may involve elements of multiple approaches to balance these considerations.

15.2 Enhanced Integration and Interoperability

A key objective for future development is to further enhance the integration and interoperability capabilities of the ADH. The development of an OpenAPI v3 web service will facilitate easier integration with diverse systems and tools, providing standardized, accessible interfaces for interacting with the data hierarchy. This web service will lower the barriers to ADH adoption and expand its utility across different platforms and environments.

The ADH aims to eliminate technical "drudge work" that currently consumes significant engineering resources. Planned capabilities include automatic units conversion, axis system transformations, table interpolation/extrapolation, support for multiple table formats, equation rollup capability, and regular expression parsing. These features will automate common data management tasks, allowing engineers to focus on high-value design and analysis activities.

Expanding the utility of the Standard Evaluator developed in CLIN-002 to provide a common interface with other workflow frameworks (such as Dagster and Airflow) and PIDO frameworks (such as Ansys ModelCenter and Dassault Process Composer) will significantly reduce re-work and enhance innovation. By making the Standard Evaluator, and by association the ADH and its integrated tools, plug-in compatible, we enable seamless integration into any workflow automation framework. This compatibility ensures that the ADH can be effortlessly incorporated into various digital ecosystems without the need for additional resources to re-integrate tools.

This approach not only minimizes the time and effort required for re-work but also fosters a more agile and innovative environment. Engineers and developers can focus on advancing their projects rather than dealing with integration challenges. The plug-in compatibility of the Standard Evaluator will streamline processes, enhance collaboration, and accelerate the adoption of new technologies. Ultimately, this will drive greater efficiency and innovation across the aerospace industry, enabling the development of cutting-edge solutions that meet the evolving demands of the sector.

15.3 Advanced Capabilities

Future development will incorporate several advanced capabilities that enhance the utility and impact of the ADH. Uncertainty quantification will be integrated into the framework, enabling more robust and reliable design decisions by accounting for variability and uncertainty in design parameters and performance predictions. This capability is essential for risk management and decision-making in complex aerospace programs.

Developing capabilities for "rollups" of mass properties, cost, and other critical metrics will facilitate comprehensive system-level assessments and analyses. These rollups aggregate component-level data to provide system-level insights, supporting holistic design optimization and trade-off analysis. For Model-Based Systems Engineers, these system-level metrics are essential for evaluating overall performance and making informed design decisions.

It is time to expand the use cases within the Aircraft Systems Design community. We should explore areas such as safety analysis, fault hazard assessments, human factors, systems integration, and predictive maintenance. By delving into these use cases, we can uncover new opportunities to enhance the design, reliability, and efficiency of aircraft systems. This exploration will not only address current challenges but also pave the way for innovative solutions that improve overall safety and performance in the aerospace industry.

15.4 AI Integration and Future-Proofing

A forward-looking aspect of the ADH development is its alignment with emerging artificial intelligence and machine learning technologies. The selection of Pydantic v2 aligns with common data management approaches in many machine learning and Generative AI frameworks, positioning the ADH for future integration with these advanced technologies.

As trust in AI matures, the ADH will be enhanced to support AI integration by implementing Model Context Protocol (MCP) Server and PydanticAI Agents. These capabilities will support advanced analytics and machine learning applications, opening new possibilities for data analysis, insight generation, and design optimization. This AI-friendliness will enhance the utility and effectiveness of the ADH, supporting innovative approaches to aircraft design and analysis.

15.5 Deliverables and Documentation

Future deliverables will include a comprehensive set of resources to support ADH adoption and implementation. Our GitHub Open Source Repository provides access to the ADH Python package, allowing developers to download, install, and integrate the framework with their own tools and processes. Comprehensive unit tests will ensure reliability and correctness, providing confidence in the framework's behavior under different conditions as it evolves.

Detailed documentation will facilitate understanding and usage of the ADH, including tutorials, examples, and reference materials. This documentation will cover not only the technical aspects of the framework but also best practices, integration strategies, and use cases, providing a comprehensive resource for ADH users. A final NASA report will summarize the project's progress, findings, and outcomes, providing a comprehensive overview of the ADH development and its potential impact.

16. The Critical Case for Continued Investment

For Model-Based Systems Engineers at NASA, the continuation and expansion of the Aircraft Data Hierarchy initiative represents a strategic investment in the future of aerospace engineering. The case for this continued investment rests on several compelling factors that highlight the transformative potential of the ADH.

16.1 Addressing Fundamental Challenges

The ADH directly addresses fundamental challenges that have long impeded efficient, effective aerospace design. The lack of standardization and interoperability in current data management approaches creates significant inefficiencies, with engineers spending excessive time on manual data translation and reconciliation rather than value-added design activities. These inefficiencies occur inside an organization and when trying to coordinate/collaborate between organizations. By providing a standardized framework for aircraft data, the ADH eliminates these inefficiencies, enabling more productive use of engineering resources.

The current disconnect between Model-Based Systems Engineering (MBSE) and Model-Based Systems Analysis (MBSA) creates barriers to implementing comprehensive, integrated design methodologies. The ADH bridges this gap, providing a unified framework that supports both systems engineering and systems analysis perspectives. This integration enables true model-based systems engineering, where models serve as the authoritative source of design information throughout the development lifecycle.

16.2 Enabling Digital Transformation

NASA, like many organizations, is pursuing the 'Digital Transformation' to enhance efficiency, agility, and innovation. The ADH is a critical enabler of this transformation, providing the data standardization and integration necessary for implementing digital engineering methodologies.

The ADH enables the concept of logging a "Digital Thread" that maintains continuity and consistency of design information throughout the product lifecycle. This "Digital Thread" enables traceability from requirements through design, analysis, manufacturing, and operations, ensuring that design intent is preserved and that changes are consistently propagated throughout all aspects of the system.

Furthermore, the ADH provides a foundation for implementing "digital twin" approaches, where virtual representations of aircraft systems can be used for simulation, analysis, and decision-making throughout the lifecycle. These digital twins enable more informed, data-driven decisions that enhance performance, reliability, and efficiency.

16.3 Competitive Advantage and Leadership

In an increasingly competitive global aerospace environment, NASA's leadership in technology development and innovation is critical for maintaining U.S. preeminence in aerospace. The ADH provides a competitive advantage by enhancing the efficiency and effectiveness of NASA's engineering processes, enabling more rapid development of innovative technologies and concepts.

By establishing an open, standardized framework for aircraft data, NASA also positions itself as a leader in aerospace data standardization, influencing industry practices and promoting interoperability across the aerospace ecosystem. This leadership role extends NASA's impact beyond its own programs, catalyzing broader improvements in aerospace engineering practices.

16.5 Building on Progress

The progress already achieved in ADH development represents a significant investment that would be leveraged by continued development. The foundational framework, architectural decisions, and prototype implementations provide a solid basis for further advancement, ensuring that continued investment builds on existing work rather than starting anew.

The engagement and collaboration established with key stakeholders, including NASA centers, other government agencies, and industry partners, represents another valuable asset that would be leveraged by continued development. This collaborative ecosystem provides diverse perspectives, requirements, and expertise that enhance the quality and utility of the ADH.

17. Conclusions and Recommendations

17.1 Conclusions

The Aircraft Data Hierarchy (ADH) represents a significant advancement in the standardization and management of aerospace data. By addressing the critical challenges of data integration, standardization, and interoperability, the ADH provides a comprehensive framework that enhances the efficiency, accuracy, and collaboration in aircraft design and analysis processes. The following key conclusions can be drawn from the development and evaluation of the ADH:

1. Standardization and Consistency: The ADH offers a standardized based framework for aircraft data management, aligned with established industry standards such as MIL-STD-881F, SAWE RP A-8, and

ANSI/AIAA-S-119-2011. This standardization ensures uniformity and clarity in data representation, reducing ambiguities and enhancing cross-disciplinary collaboration.

2. Interoperability and Flexibility: The ADH promotes interoperability across different systems, tools, and platforms by adopting vendor-neutral standards and supporting multiple data formats (JSON, YAML, XML). This flexibility allows for seamless integration into diverse technological environments and accommodates evolving design methodologies and technologies.
3. Enhanced Efficiency and Reliability: By providing a centralized, authoritative data source, the ADH eliminates data duplication, reduces manual data translation, and enhances data integrity. This leads to more efficient design processes, reduced development timelines, and improved reliability and accuracy of design outcomes.
4. Comprehensive Data Representation: The Aircraft Digital Harness (ADH) organizes aircraft system data hierarchically, adhering to Model-Based System-of-Systems Architecture (MSoSA) guidelines, for a complete and integrated view. This structure supports engineering activities from conceptual design to detailed analysis within both Model-Based System Analysis (MBSA) and Model-Based System Engineering (MBSE).
5. Collaborative Development and Industry Involvement: The collaborative development model, involving key stakeholders from NASA, Boeing, GE Aerospace, Collins Aerospace, and the University of Michigan, ensures that the ADH addresses diverse needs and leverages industry expertise. This collaboration promotes adoption and integration of the ADH across the aerospace community.

17.2 Recommendations

To fully realize the transformative potential of the ADH and ensure its successful implementation and adoption, the following recommendations are proposed:

1. Continued Investment and Development: Continued investment in the development of the ADH is essential to build on the progress achieved and address remaining challenges. This includes detailing the individual data objects, enhancing integration and interoperability capabilities, and incorporating advanced features such as uncertainty quantification and AI integration.
2. Open Source Collaboration: Publishing the ADH on a public GitHub repository and adopting an open-source development model will facilitate broader industry participation, enhance the quality of the framework, and promote widespread adoption. This collaborative approach will leverage the collective expertise of the aerospace community and ensure the ADH remains relevant and innovative.
3. Enhanced Integration with Existing Tools: Further integration with existing MBSE and MBSA tools, such as MagicDraw and OpenMDAO, is critical for seamless adoption. Developing standardized APIs and web services will lower the barriers to integration and expand the utility of the ADH across different platforms and environments.
4. Comprehensive Training Documentation and Examples: Providing detailed training documentation, examples, and training resources will support the adoption and effective use of the ADH. This includes a variety of aircraft types, reference materials, and best practices for integration and usage.
5. Engagement with Additional Stakeholders: Expanding collaboration to include additional universities, defense organizations, and aircraft manufacturers will ensure the ADH addresses a broader range of requirements and use cases. This engagement will enhance the framework's versatility and promote its adoption across different sectors of the aerospace industry.
6. Focus on Digital Transformation: Aligning the ADH with digital transformation initiatives, such as digital twins and digital threads, will enhance its strategic value and impact. This includes developing capabilities for real-time data alignment, advanced analytics, and predictive maintenance.

By implementing these recommendations, the ADH can become a fundamental component of future aircraft design and analysis processes, driving innovation, efficiency, and collaboration in the aerospace industry. The successful development and adoption of the ADH will establish a lasting legacy, positioning NASA and its partners at the forefront of aerospace data standardization and digital transformation.

The continuation and expansion of the ADH initiative represent a strategic investment in the future of aerospace engineering at NASA. By providing the data standardization and integration necessary for digital transformation, the ADH enables more efficient, effective, and innovative approaches to aircraft design. This enhanced capability will accelerate the development of new aerospace technologies and concepts, maintaining NASA's leadership in aerospace innovation and advancing its mission objectives.

The Aircraft Data Hierarchy is not merely a technical solution but a transformative framework that will reshape how aerospace engineering is conducted. Its successful development and implementation will leave a lasting legacy, establishing a foundation for aerospace data standardization that will benefit NASA and the aerospace industry for years to come. For Model-Based Systems Engineers at NASA, this represents an unprecedented opportunity to advance the state of the art in aerospace engineering and enable the next generation of aerospace innovation.

18. References

Standards

1. MIL-STD-881F, "Work Breakdown Structures for Defense Materiel Items," U.S. Department of Defense, 2018.
2. NASA, "NASA Systems Engineering Handbook," NASA/SP-2016-6105 Rev 2, 2016. https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook
3. NASA, "Model-Based Systems Engineering in Support of Complex Systems Development," NASA/CR-2016-219333, 2016.
4. ANSI/AIAA, "Flight Dynamics Model Exchange Standard," ANSI/AIAA-S-119-2011, 2011.
5. Society of Allied Weight Engineers, "Weight and Balance Reporting Forms for Aircraft (Military and Commercial)," SAWE RP A-8, 2015.
6. Air Transport Association, "E-Business Specification for Materiels Management," ATA SPEC 2000, 2015.

Common Parametric Aircraft Configuration Schema (CPACS) API

7. DLR-SL, "CPACS," available at: https://github.com/DLR-SL/CPACS.
8. NASA, "Aviation Safety Program," NASA, 2020. https://www.nasa.gov/aeroresearch/aviation-safety-program/
9. Daniel de Vries, "Openlego 1.5.0," available at: https://github.com/daniel-de-vries/OpenLEGO.
10. "Python CPACS Toolbox: A Python toolbox for reading and writing CPACS data files," 2020.
11. "cpacspy 0.1.8," available at: https://github.com/cfsengineering/cpacspy.
12. "CPACS Library," 2020.
13. "OpenCPACS: An open-source CPACS reader and writer written in C++," 2020.
14. Ciampa, P., et al., "Towards the Third Generation MDO Collaboration Environment," AGILE, 2016.
15. "CPACS: Recent Advances in Establishing a Common Language for Aircraft Design," 2020.

Aircraft Data Markup Language (ADML)

16. "ADML Website," available at: https://vtechworks.lib.vt.edu/bitstream/handle/10919/24826/admlJAIS13.pdf;sequence=1.

Aviary

17. Gratz, E., et al. (2024). Aviary: An Open-Source Multidisciplinary Design, Analysis, and Optimization Tool for Modeling Aircraft. *OpenMDAO Aviary Documentation*. Retrieved from https://openmdao.github.io/Aviary/intro.html

OpenMDAO

18. Gray, J. S., Hwang, J. T., Martins, J. R., Moore, K. T., and Naylor, B. A., "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," Structural and Multidisciplinary Optimization, vol. 59, no. 4, pp. 1075-1104, April 2019.