



Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com



A methodology for constructing the aircraft design schema

Xiaojian SUN, Mingqiang LUO *, Zhiyang CUI, Zhenpeng JI, Hu LIU

School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China

Received 2 July 2022; revised 23 September 2022; accepted 17 October 2022

Available online 8 March 2023

KEYWORDS

Aircraft collaboration design;
Aircraft design schema;
Conceptual design;
Digital engineering;
Schema construction methodology

Abstract Aircraft collaboration design is based on a unified set of schemas containing data and model representations and interfaces across disciplines. The aircraft design schema is a central data model that specifies the paradigm for the representation of a full lifecycle model of aircraft design. The construction of the schema currently lacks theoretical guidance. This paper designs a novel methodology to construct the aircraft design schema, a fully functional, logical, and self-consistent methodology. The methodology defines a schema matrix that includes implementation strategies, principles, processes, meta-object facility, views, scenarios, and products. The aircraft conceptual design schema is constructed according to the methodology, which is a hands-on approach to help understand the concept and implementation. The features of the methodology are analyzed, and the methodology and the aircraft conceptual design schema are presented to address the pain points of multiple solution trade-offs, multiple iterations, and multidisciplinary coupling in aircraft conceptual collaboration design.

© 2023 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Aircraft collaborative design refers to the timely sharing of information in the aircraft design process to leverage the necessary tools, skills, and workforce to facilitate interdisciplinary collaboration. The key to the collaborative design is to share information promptly and ensure that participants access the information they need. The traditional document-based collaborative design uses printed or electronic documents to share information, which is not delivered promptly and requires the recipient to deconstruct the document to access the information. The model-based collaborative design uses models as

the carrier for sharing information. Models and software tools are naturally self-consistent and facilitate the timely sharing of information. One of the key issues in moving from a document-based design approach to a model-based digital engineering ecosystem is the management of models. At the same time, collaborative design suffers from the painful problems of multiple solution trade-offs, multiple iterations, and multidisciplinary coupling. The leading cause of these problems is the coarse granularity of the management of aircraft design models and the lack of a unified representation. Therefore, collaborative design requires a unified model representation paradigm that can act as a model interface for collaborative design and manage models from different granularities.

* Corresponding author.

E-mail address: luomingqiang_buaa@163.com (M. LUO).

<https://doi.org/10.1016/j.cja.2023.03.009>

1000-9361 © 2023 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

The aircraft design schema is a central data model¹ that specifies the paradigm for the representation of a full lifecycle model of aircraft design. The schema is an abstract collection of metamodels that identifies the types of design information, the relationship of those elements, and the extensibility for including new and additional types of design information.² The schema defines the representation of all possible uses of models in aircraft design, including the definition of basic concepts, models, and design results, in different life cycles and professional contexts. The schema is a standard model specification, which only specifies the representation of the model, not the modeling specification or the modeling guidelines. At the same time, the schema is also a standard format for exchanging information for collaborative design model exchange, which helps solve the problem of collaborative aircraft design by managing the model in a granular way and specifying a uniform model representation.

In 2018, the US Department of Defense presented its digital engineering strategy, whose first goal is to “Formalize the development, integration, and use of models to inform enterprise and program decision making”.³ By defining a model-based approach and process, it is possible to better capture the “as-is” and “to-be” engineering enterprise architecture and improve the speed and quality of engineering and decision processes.⁴ The construction of an aircraft design schema is an integral part of formal modeling. An Authoritative Source of Truth (ASoT) is the second goal of the digital engineering strategy. The ASoT architecture provides the framework to ensure the validity and truthfulness of digital artifacts as they are managed throughout their lifecycle and the integrity of the enabling technological systems.⁵ By constructing an ASoT for civil aircraft pre-research, Sun et al. found that the aircraft design schema is the basis for the construction of ASoT and that without a unified standard model, the ASoT will degenerate into systems of record.⁶ The unified standard model is based on the aircraft design schema.

However, the construction of the schema currently lacks theoretical guidance. Ontologies in some specialisms are constructed, where Airbus builds a preliminary ontology definition for aerospace assembly lines.⁷ Systems Modeling Language (SysML) is a modeling language of Model-Based Systems Engineering (MBSE).⁸ The National Aeronautics and Space Administration has successfully applied the SysML in several projects.^{9–12} However, the domain ontology represented by the SysML not only defines the model representation effect but also specifies the modeling specification and modeling guidelines, which is more complex, lacks flexibility, and has a long training cycle. They do not have cross-disciplinary domain implementation, and are more challenging to implement. In order to promote interdisciplinary cooperation between research institutes, the German Aerospace Center has been building the Common Parametric Aircraft Configuration Schema (CPACS) since 2005.¹ CPACS is implemented as a schema definition for the extensible markup language. CPACS defines the elements, attributes, and structure of all information that may be used in the conceptual design of an aircraft. It allows for the effective linking of any interdisciplinary analysis capabilities.

This paper, first, designs a novel methodology to construct the aircraft design schema, including implementation strategies, frameworks, principles, processes, Meta-Object Facility (MOF), views, scenarios, and products. The methodology is

a fully functional, logical, and self-consistent methodology that covers multiple granularities and helps users construct a coherent and evolving aircraft design schema (Section 2). Second, the application of the methodology to the aircraft conceptual design phase, the Aircraft Conceptual Design Schema (ACDS), and the hands-on approach help users understand the concept and implementation of the methodology (Section 3). Finally, the features of the methodology and ACDS are then analyzed, and the methodology is clearly presented to address the pain points of multiple solution trade-offs, multiple iterations, and multidisciplinary coupling in aircraft design (Section 4).

The methodology and the ACDS built through the methodology are the basis for the application of the aircraft design solution in the ASoT. A collaborative design method for civil aircraft based on an enduring and evolutionary ASoT in ICAS 2022 is designed.¹³ In this paper, Sun et al. designed a human-software-hardware collaborative design method based on the ASoT and ACDS. ASoT and its derivative systems also were designed and developed. A concrete example of the future civil aircraft was developed by applying the ASoT and its derivative systems.

To summarize, the contributions of the proposed framework are presented as follows:

- (1) A methodology for constructing the aircraft design schema. The methodology can help system engineers, system analysts, and discipline engineers construct the schema.
- (2) A hands-on practice of the methodology using an aircraft conceptual design example, which is aircraft conceptual design schema, to help users understand the detailed operation of the methodology and make it more implementable.

2. Methodology

As complex equipment, aircraft cover many design phases, disciplines, and stakeholders. Aircraft data and models are generally expressed in an agreed-upon format. Such expressions may occur occasionally and are prone to untimely communication, resulting in poor information interaction between disciplines. The conventional representation of data and models is poorly coupled with design tools, and reuse is difficult. Therefore, a comprehensive methodology is needed to form an aircraft data and model schema that can be applied to engineering, improve model consistency, and reduce engineers' burden, thus supporting digital transformation and digital engineering ecosystem construction.

The methodology designed in this paper contains a set of concepts and implementation strategies and processes for constructing schema, and the products that result from using the process. Specifically, this methodology is well-structured and provides a set of guidelines. This methodology contains a set of concepts: schema matrix, implementation strategies, MOF of schema, a process for constructing a schema, views, scenarios, and products that result from the process.

In general, the schema is constructed by the system engineer using the methodology, with the cooperation and participation of the system analyst and discipline engineers, and guidance

and direction from the project manager. The system engineer needs to communicate closely and iterate with the system analysts and discipline engineers, and the project manager needs to drive the implementation from the top down when necessary.

Schema covers the expression of data and models in the scope of the System of Interest (SoI) and provides a standardized data interface and model expression that can be easily used and communicated by various engineers.

This methodology is novelty in that no methodology for constructing aircraft design schema has been found after a literature review. The only schema in aircraft design is CPACS. Nagel et al. described the objectives and means of establishing CPACS,¹ but did not address how it is established, i.e., the methodology.

2.1. Schema matrix

The schema matrix (Table 1) is a construction framework that can be represented as a Zachman style matrix.¹⁴ The rows are the stages of the whole life cycle of SoI and the stages of the system engineering “V” model.^{15,16} Taking civil aircraft as SoI, rows contain the phases of requirements analysis and concept definition, product and service definition, manufacturing certification, and industrialization. The requirements analysis and concept definition phase is the process of gradually forming a feasible product concept plan and initiating the project. It includes the process of starting from market and business opportunity analysis, conceptualization, and conception to generate aircraft and service product solutions, the economic and technical feasibility analysis of the solution, and finally, the formal formation of the project, in which the basic product concept and feasibility plan are formed. It can be subdivided into three sub-stages: concept development, project demonstration, and feasibility study. After the project is established and feasibility is approved, the whole development process, from the start of aircraft development to the final formation of aircraft products, is the process of developing a product system to meet customer needs. The product and service definition stage is located on the left side of this “V” development process, mainly based on the definition of requirements and design decomposition of the concept plan, and finally, complete the detailed design process of aircraft products and services. It can be divided into two sub-stages: the preliminary design stage and the detailed design stage. The manufacturing certification phase is located on the right side of the “V” development phase, which is the process of manufacturing, integration, verification, and product confirmation of the aircraft product, and finally forming the aircraft product and completing the delivery of the first aircraft or the first batch of aircraft. Specifically, it can be divided into two sub-stages: the full-scale trial production stage and the test flight certification stage. The industrialization stage is the completion of product development, product and service improvement according to the operation, completion of product and service confirmation, and final project acceptance. Meanwhile, the product is transferred to the batch production stage. Production is carried out according to market orders, usage improvements are performed as needed, and product support and customer service work is carried out to realize scale and industrialization gradually. As time progresses, the model decommissioning process is carried out according to actual conditions. It can be divided

Table 1 Schema matrix.

Stage	Principle	MOF	Process	View	Scenario	Product	Sub-stage
Requirements analysis and concept definition	Construction principles of schema	MOF of requirements analysis and concept definition stage	Construction process of schema	Views of requirements analysis and concept definition stage	Scenarios of requirements analysis and concept definition stage	Products of requirements analysis and concept definition stage	Concept development / Project justification / Feasibility study
Product and service definition		MOF of product and service definition stage		Views of product and service definition stage	Scenarios of product and service definition stage	Products of product and service definition stage	Preliminary design / Detailed design
Manufacturing and certification		MOF of manufacturing and certification stage		Views of manufacturing and certification stage	Scenarios of manufacturing and certification stage	Products of manufacturing and certification stage	Full-scale trial production / Test flight certification
Industrialization		MOF of industrialization stage		Views of industrialization stage	Scenarios of industrialization stage	Products of industrialization stage	Product and service acceptance / Continuing operations / Retirement

into two sub-stages: the product and service acceptance stage and the continuous operation stage.

The columns are pillars that help system engineers think in multiple dimensions. Pillars contain principle, MOF, process, view, scenario, and product. The six pillars contain the temporal and spatial, dynamic and static, conceptual and process dimensions of schema construction, and they encompass all the issues to be dealt with throughout the process of constructing a schema. Principles mainly describe the basic principles followed in the schema construction process. MOF defines the data and model connotations of the schema in four layers: meta metamodel, metamodel, conceptual model, and instance model. The process describes how to construct a schema in each phase of the SoI lifecycle, providing a generalized and actionable process that system engineers can rely on to construct a schema step by step. Views describe the stakeholders associated with the schema and their views, deconstructing the schema of the SoI into multiple views through the stakeholder perspective. Scenarios help clarify the scenarios in which the schema is involved, i.e., the way, time, stage, and specifics of the schema that is used and the role of the schema in the design. The products are the result of each phase of the process. In these six pillars, the principle serves as a guide, the MOF is the basis of expression, the process is a practical guide, the view and scenario can assist the process in identifying data and models, and the products are the results.

Each cell at the intersection of the rows and columns represents a specific operation or action performed at each step in the schema matrix. By executing the schema matrix, system engineers can clarify the connotation of the cells corresponding to each phase in the SoI lifecycle, including why the schema of this phase should be constructed, what specific behaviors are needed to construct the schema, what is the product of the completed construction, and what is the relationship between the behaviors of this phase and other phases.

2.2. Implementation strategies

The purpose of the implementation strategies is to help system engineers make appropriate use of the methodology when constructing the schema. Developing schema adds workload to engineers, and the upfront use of schema by engineers may also add workload, so there is a need to help engineers reduce their workload in other ways. We have developed seven strategies based on this idea:

- (1) Columns: Columns are not sequential but should start with the most accessible category and avoid deleting any columns. Columns are coupled and interact with each other. Each column describes a unique perspective that needs to be considered holistically and avoids redundancy.
- (2) Rows: Rows are the sequence of the full lifecycle development of the SoI and need to be arranged from top to bottom. Rows can be refined, disassembled, and tailored during the lifecycle stages to form a complete representation of the specific stage model.
- (3) Enduring: Due to the enduring nature of the SoI lifecycle development, the schema between rows should have the characteristic of enduring, which can accommodate the expression of the data and models of the previous

row (previous stage) or take over its data and models smoothly. The enduring characteristic is reflected in the models at each stage of the lifecycle development, not starting from scratch but continuing to be developed based on the models of the previous stage. Although the expressions of the models at each stage are very different, in essence, they are all products of a specific perspective at a specific stage of interest system development and are capable of expressing a specific aspect of the interest system.

- (4) Record: Avoid modifying the names of rows and columns. If system engineer does need to modify them, they should record and publish them to avoid confusion about their meaning.
- (5) Recursive and generic: The methodology is recursive and generic, meaning that it can be used to analyze any system of interest at any life cycle stage. It needs to be clear that establishing goals and boundaries of SoI depends on the system engineer and that these decisions can significantly impact the framework and the results. The recursive and generic nature means that the framework's transverse rows are not set in stone and are bounded by a particular stage of the SoI, which can be internally divided into sub-stages, and the transverse rows are those sub-stages.
- (6) From small to large: The schema should be constructed from small to large, first in a small-scale pilot, to gain recognition and experience, gradually improve, then form a specific ecological foundation, and finally promote the construction and application on a large scale.
- (7) Continuous maintenance: Schema construction is not an overnight process. It requires continuous iteration and optimization and is a long-term process that will eventually form a culture of digital transformation. Stakeholders need to be involved and contribute to the formation and maintenance of the schema.

2.3. Principles

The principles guide the whole process of construction, in which five principles need to be followed.

- (1) Foundation: Based on the existing documents and data. Schema is not an airy building but based on previous experience. The schema is an optimization and upgrading of existing workflow and working methods, so it must be based on existing documents and data.
- (2) Deliverables: Including the deliverables of each discipline and the SoI. Deliverables are the products that can express the more comprehensive SoI hammered out in engineering practice, and supporting delivery is one of the purposes of schema.
- (3) Transmission. One of the purposes of schema is to enhance the efficiency between disciplines. Therefore, the transmitted data should be used as an essential reference. The current data should be communicated between disciplines in a more uniform format. Using the passed data as a basis can reduce the engineer's workload and cause as little disruption as possible to the existing work mechanism.

- (4) Tools: Focus on the data generated by the design tools. The tools are carefully selected and are the essential condition for generating data. The data generated by tools have conformed to some of the characteristics of the schema. Some of the tools may have been secondarily developed and can form data more aligned with the schema features. Based on the formed schema, secondary development of the tools may also be required to facilitate the tool to match the schema directly.
- (5) Standards: Follow the current national, industry, corporate standards, and airworthiness standards. The schema aligned with the standards can be more conducive to implementation and draws on the experience accumulated during the standards development.

2.4. MOF

MOF^{17–19} is related to the schema construction process. The enduring model lifecycle allows system engineering to return from any state to another state if some definition is missed or incorrect. The MOF allows models to be added, removed, and modified at any point in the model lifecycle.

MOF is a four-layer framework as shown in Section 2.1, i.e., (from top to bottom) the meta metamodel layer, meta-model layer, conceptual model layer, and instance model layer. The definition of metamodel, conceptual model, and instance model are not provided in the methodology. Because these three layers of models are closely related to the SoI, they need to be defined during the practice of methodology.

2.4.1. M3: Meta metamodel layer

The meta metamodel is the language used to construct the metamodel and is the most abstract model. This methodology defines meta metamodels as “element” and “relationship”. The basis of all data and models can be attributed to the combination of “element” and “relationship”. The “element” is the most basic model that can exist on its own and can become a model on its own, or it can form a model together with other “element” through “relationship”. The “relationship” is the abstract language expressing data and model relationships.

2.4.2. M2: Metamodel layer

The metamodel defines the specification of the model, is the abstraction of the conceptual model and is the instantiation of the meta metamodel. The design process of the metamodel is shown as follows:

- (1) Instantiate “element” and define elements. Elements are used to describe the basic features of a conceptual model, and a class of conceptual models with intrinsic meaning can be resolved into a basic metamodel. Element names, attributes, and other feature values need to be defined.
- (2) Instantiate “relationship” and define relationships. The decoupling conceptual model analyzes the relationships and defines their names. Relationships support joining two or more elements.

The metamodel layer is the key to reflecting the characteristics of enduring. Different stages of the conceptual model

may have the same metamodel, and the combination of finite metamodels can express the conceptual model and realize the coherence and evolution of the conceptual model between stages.

2.4.3. M1: Conceptual model layer

A conceptual model is an instantiation of a metamodel and an assemblage of elements and relationships. Different elements and relationships are combined in a specific way and supplemented by the requirements of the actual scenario to form a conceptual model. A conceptual model is an “abstract language” defined to describe an instance model with the attributes of the instance model but without the data associated with the instance and can be instantiated as a series of different instance models.

2.4.4. M0: Instance model layer

Instance models are constructed by engineers and are human-readable and understandable models. The conceptual model is instantiated by adding data and models related to the SoI. The conceptual model can be instantiated into different instance models for different SoI or solutions. Different products have different instance models.

Defining the MOF with its four layers of model definition involves the issue of model granularity. Model granularity is related to the accuracy of model representation and ease of implementation (Fig. 1). The finer the model granularity is, the higher the model expression accuracy is, and the more favorable it is to express the SoI. Still, the implementation difficulty becomes higher, and engineers must spend more time adapting to the schema. The learning cycle increases with the model granularity, which is not conducive to forming a culture of digital transformation. Therefore, it is suggested that we follow the construction principles, clarify the existing foundation, refer to the deliverables, tools, and standards, implement in steps, form a preliminary version first, and continuously iterate and optimize to finalize the granularity of model expression.

2.5. Process

The process can clarify the concrete, actionable steps of schema construction. Referring to the ontology constructing process,²⁰ the methodology identifies a series of activities to be completed during schema development: acquiring data, conceptualizing, normalizing, integrating, evaluating, implementing, documenting, and maintaining. According to the activities, the data and models’ life goes through the following states: conceptualization, normalization, integration, instantia-

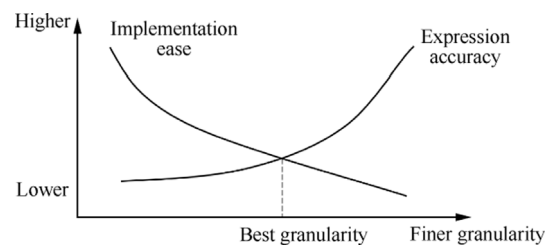


Fig. 1 Expression accuracy and implementation ease related to model granularity.

tion, and maintenance. Moreover, most states' supporting activities are acquiring data, evaluating, implementing, and documenting.

2.5.1. *Acquiring data*

The purpose of acquiring data and models is to obtain instance models. According to the schema construction principle, the source of schema can be the document report, historical report, deliverables of each discipline, the result generated by design tools, the result of technical research, even experts, books, manuals, figures, and tables. Analyzing the sources makes it possible to clarify the data and models contained therein.

The role of the instance model is to express a specific feature of the SoI, and one or more sets of instance models are needed to express the complete picture of the SoI at the specific stage. In general, the instance models of SoI have a tree structure. Combining the following techniques can obtain data and models: brainstorming, interviews, and formal and informal textual analysis. If there is no clear idea about the model, these techniques can help clarify the first set of data and the model.

Unstructured interviews with experts can help make sense of the tree-structured data and models. Textual analysis can be conducted using both formal and informal books and manuals to refine the data and models and their meaning. Informal textual analysis can examine the main concepts given in books and manuals. Formal textual analysis can determine the structure of the data and model tree and the kind of knowledge (concepts, attributes, and relationships) contributed by each structure. Structured interviews with experts can obtain specific knowledge about concepts, attributes, and relationships, evaluate the conceptual model after the conceptualization activity is completed, and assess the implementation.

2.5.2. *Conceptualizing*

Conceptualize the data by extracting the model and data from the source data. This behavior corresponds to the extraction of conceptual models from instance models in the MOF.

The normalization models are already available to help with conceptualization. The system engineer must identify and collect all valuable and potentially helpful domain data and models and their meanings. Once the conceptualization is completed, it is crucial to be able to specify the data, models, and their attributes and instances. Each data or model has its name, type, remarks, and other attribute values, and can also be instantiated as data and models of specific objects.

2.5.3. *Normalizing*

Standardize the conceptualized model to produce standardized data and model representations. The metamodel is extracted from the conceptualized model by normalizing.

The purpose of normalization is to form a semi-formal or formal representation of the model normalization in a natural language, semi-formal or formal form. It is recommended to consider at least the following two points: (A) The purpose of the modeling. The purpose includes the model's intended use, usage scenarios, and end users; (B) The level of formality of the model. It depends on the level of formality that will be used to codify the terms and their meanings. The level of normalization can be classified as highly informal, semi-informal, semi-formal, or strictly formal models. The level of

normalization depends on whether the models and their meanings are measured in a language between natural and strictly formal language.

The intermediate elimination method determines the normative model instead of the traditional top-down or bottom-up approach. A bottom-up approach lacks a unified model consideration and tends to create conflicts in key terms among different professions and departments. Similarly, the top-down approach relies on top-level consideration, making it difficult to standardize the model and challenging to implement a model that is out of line with actual needs. The intermediate elimination method allows the systems engineer to define the main concepts of the models. After agreeing on these models and their definitions, the system engineer can continue to specialize or generalize them when necessary. Since the complete completeness of the model specification document cannot be proven (at any time, someone may find a new related model to be included), it is crucial to ensure that the following properties are present in a good schema: conciseness, partial completeness, and consistency.

2.5.4. *Integrating*

Compare the normalized conceptual models, unify the duplicated models, and carry out entity disambiguation and coreference resolution.

Entity disambiguation refers to differentiating different models with the same name, and coreference resolution refers to merging expressions of the same model with different representations. To speed up schema construction, we consider reusing schema already constructed in other domains instead of starting from scratch.

- (1) Check the metamodel to select those conceptualization models more appropriate for the SoI and disciplines. The object is to ensure that the new and reused definitions are based on the same basic set of terms. If the existing metamodel is unsuitable, a new metamodel should be defined and implemented in formal language.
- (2) Whether the system engineers reuse an existing metamodel, the next step is determining which model libraries provide definitions and implementations of terms consistent with the terms identified in the conceptualization. Once the most appropriate terms have been selected, if the metamodel on which these terms are based is different from the metamodel system engineer used to build the model, the system engineer should check if a translator is available to convert the definitions into the target language.

Sometimes, a term in model (e.g., centimeter, foot) may be included in a given model, but the conceptual model does not provide it. In this case, the need to include the missing definition should be justified, and the benefits of such inclusion should be explained to the model maintainer.

The result of the integrating activity is a preliminary conceptual model and meta-model; for each of them, the terms will be used for their definition.

2.5.5. *Evaluating*

Evaluate if the model is usable and if unavailable models need to be iterated for renormalization, integration, or new designs. Evaluating refers to the schema-based technical judgment of

the data and models and its software environment at each stage of the model lifecycle and between stages. Evaluating encompasses verification and validation. Verification refers to the technical process of assuring the correctness of the data and models, its associated software environment, and documentation relative to the schema at each stage of its lifecycle. Validation ensures that the model, software environment, and documentation correspond to the system that they are supposed to represent.

Evaluation can generally be done in centralized review and trial feedback. The centralized review relies on the knowledge base and experience of experts involved in the model lifecycle, and focuses on evaluating the usability of the model, then modifying the controversial model and determining the model's compatibility in the next phase. The trial feedback is sent to the engineers for trial, and the engineers use the conceptual model to form an instance model and then evaluate the change in the engineer's habits, the change in the existing tools and systems, the improvement in the consistency of the model, and the reduction in the workload. Feedback is obtained, and the conceptual model is modified.

The output of this activity is an evaluation and feedback document and a revised conceptual model. The output should include a description of how the conceptual model was evaluated, the techniques used, the types of errors found in each activity, and the sources of knowledge used in the evaluation.

2.5.6. Implementing

The conceptual model will be distributed to system analysts or discipline engineers for practical use. The system analyst or discipline engineer can use the conceptual model to generate large-scale instance models for testing.

The implementation of conceptual models requires the use of an environment that supports both metamodels and conceptual models. This environment includes an ASoT that can support the schema's data and model management, usage, and instantiation, and a set of design aids, which is a close connection to the ASoT, to help engineers reduce the model generation and interaction workload.

When the implementation is complete, the system analyst or discipline engineer needs to form feedback on the implementation to help iterate and optimize the schema.

2.5.7. Documenting

The system engineer documents throughout the process for rollout or continued updates after a period, laying the foundation for a sound ecology. The methodology requires documentation as an activity throughout the schema development process, and documents must be generated at each phase's completion.

2.5.8. Maintaining

Regularly maintain the schema, updating and modifying it when appropriate. Maintenance activities are carried out throughout the entire lifecycle of the schema and are an essential part of building a sound ecology.

2.6. Views

The methodology argues that it is necessary to clarify which stakeholders decompose data and models into which views at

each stage of the SoI lifecycle. After forming the views, it is necessary to clarify what data and models are included in these views and how they are related to the constructed schema. Clarifying the above issues will help us develop the ability to generate views automatically and highlight the importance of data and models. The overlap among different views is the relatively important model and data. Defining data and models after defining views can help clarify the granularity of data and model representation.

2.7. Scenarios

In the SoI lifecycle, data and models are used in very different scenarios, the essence of which lies in the fact that data and models are very different at each stage of the lifecycle. Identifying the importance and granularity of data and models is beneficial by sorting out and analyzing typical scenarios and clarifying what role they can play in which situations.

2.8. Products

The schema construction requires a set of objective and visible products. In general, the methodology recommends at least the following products:

- (1) A schema that covers the various lifecycle disciplines within the boundaries of the SoI during this construction, providing a standardized representation. (A) Terminologies, meta-models; (B) Conceptual models; (C) Instance models for a particular object. The instance models can be used as the best practices to help other engineers understand and use the schema.
- (2) A set of documents. The schema construction process requires a set of documented documents. After acquiring data, a data model source document is formed; after conceptualizing, a conceptual model document is formed; after normalizing, a normalization document is formed; after integration, an integration document is formed; after evaluating, an evaluation document is formed; after implementing, an implementation document is formed; after maintaining, a maintenance document is formed. Finally, it is also necessary to precipitate the schema constructing process as enterprise knowledge accumulates.
- (3) An ASoT and a suite of design aids. The ASoT needs to be able to support the use and instantiation of the conceptual model fully. The storage and communication of the models are based on the ASoT. At a minimum, the ASoT should provide parser tools for meta metamodels, metamodels, conceptual models, and instance models to ensure that conceptual models and instance models can be managed; a translator to ensure portability of models in other target languages; an editor to add, delete, or modify models; a browser to inspect and browse conceptual models and instance models; a searcher to find the most appropriate models; an evaluator to detect incomplete, inconsistent, and redundant models; an automatic maintainer to manage the inclusion, deletion, or modification of the existing models. A suite of design aids includes secondary development tools or interfaces for parts of the software. These tools or interfaces can be

adapted to the schema and help with conceptual models and generating instance models. Engineers need simple training to learn to use them and do not need to spend extra effort focusing on the model format and interface. Compared to the model generation process in document-based design, these design aids can reduce the workload for engineers.

There are similarities in the design process of aircraft and other complex equipment, so this methodology appears to be applicable to other engineering systems. The design schema for other complex equipment can be modified using this methodology. However, the stage in schema matrix (Table 1) is based on aircraft design process. The MOF is designed for the data and model in aircraft design. The Process is extracted from the aircraft design process.

3. Case study

This paper focuses on the aircraft conceptual design phase and helps engineers understand the details and products of the schema construction methodology by constructing an ACDS, and this practice can also be used as a case study for engineers' reference.

3.1. Practice of implementation strategies and ACDS matrix

According to the "COMAC systems engineering manual" (4th ed),¹⁶ The "X" model of COMAC system engineering divides the whole life cycle of a product (an aircraft) into requirements analysis and concept definition, product and service definition, manufacturing certification, industrialization, and disposal. The requirements analysis and concept definition phase includes the concept development, project demonstration, and feasibility phases. There is a project decision gate between the project demonstration and feasibility stages. In order to meet the requirements of this decision gate, the work to be completed in the "project demonstration" phase is shown as follows:

- (1) Comprehensively analyze market segments, competitive products, and technology developments, and synthesize the needs of potential integrated customers to propose preliminary commercial aircraft system-level commercial requirements and objectives.
- (2) Determine the commercial aircraft product system development objectives and major technical specifications.
- (3) Propose multiple concept solutions to meet commercial aircraft product system commercial requirements and objectives, and conduct trade-offs.
- (4) Conduct a technology maturity assessment, conduct a pre-study of crucial technologies, and identify critical technologies and solutions for the project.
- (5) Prepare preliminary work breakdown structure and project milestone plan, and complete project proposal.
- (6) Propose the total project development budget and stage investment intensity and set the cost target for individual machines.
- (7) Decompose the project feasibility phase funding.

This implementation aims to see the big picture in a small way. The SoI and the phases are the conceptual design phase of the aircraft requirements analysis and concept definition phase, which is one of the tasks that should be completed in the project justification phase to propose one or more conceptual solutions that meet the commercial needs and objectives of the commercial aircraft product system. The elements included are requirements analysis and architecture design, initial sizing and conceptual sketch, configuration layout, overall arrangement, propulsion, aerodynamics, weight, performance, and cost analysis.^{21,22} The final design outcome of the conceptual design phase is an aircraft conceptual design solution that meets the commercial requirements of the commercial aircraft product system, which an instance model of schema can express.

Based on the schema matrix (Section 2.1) and the various sub-stages of the conceptual design, the ACDS matrix (Table 2) can be obtained.

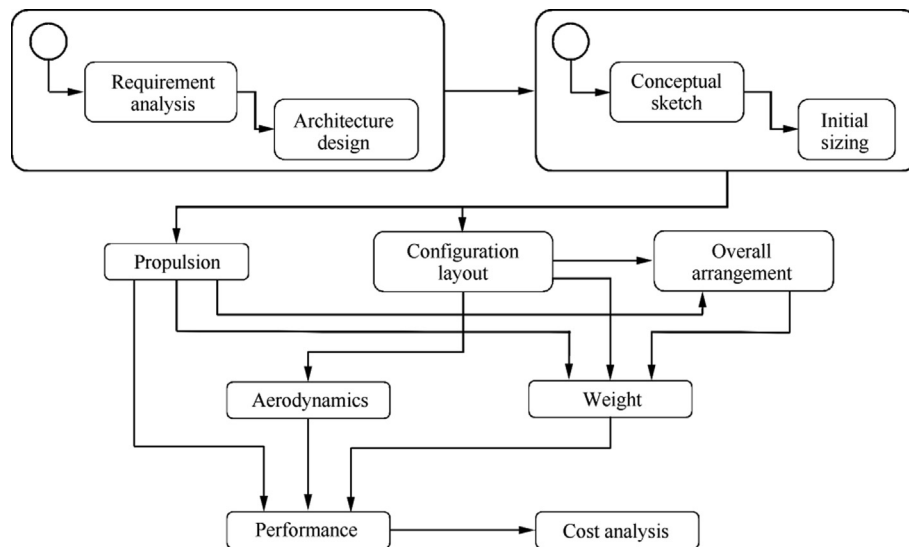
3.2. Principles and their supplements

According to the schema construction principle, the construction principle of ACDS can be determined as follows.

- (1) Foundation. The existing "Conceptual Design Report" and the original data from each discipline forming the report are used as the basis. Each discipline will also form its reports during the conceptual design phase, and these reports and raw data are also part of the foundation.
- (2) Deliverables. Sort out the current deliverables of each discipline in the aircraft conceptual design process. Generally, the deliverables are requirement set, functional architecture analysis model, initial sizing parameter table, conceptual sketches, propulsion table, configuration layout parameter table, three-view drawing, overall arrangement table, aerodynamic number table, weight calculation table, performance number table, and cost analysis parameters.
- (3) Transmission. Focus on transferring data between disciplines. The aircraft conceptual design process has more coupled relationships between disciplines. The data transfer between disciplines can be referred to in Fig. 2, in which the arrows indicate the data transfer relationship.
- (4) Tools. The tools used in the aircraft conceptual design process are requirements management tools, System functional architecture analysis (MBSE) tools, excel, CAD (Computer Aided Design) tools, CFD (Computational Fluid Dynamics) tools, Matlab, and other scientific computing tools. Some of the tools have been secondary developed, and the original data generated by the tools have been modified.
- (5) Standards. The standards to be considered for aircraft conceptual design are CCAR-25-R4 "Airworthiness Standards: Transport Category Airplanes", "Aircraft Design Manual" series, "Aircraft Flight Performance Calculation Manual", and "Civil Aircraft General Design".

Table 2 ACDS matrix.

Stage	Sub-stage	Principle	MOF	Process	View	Scenario	Product
Aircraft conceptual design	Requirement analysis and architecture design	Construction principles of ACDS	MOF of requirement analysis and architecture design	Construction process of ACDS	Views of requirement analysis and architecture design	Scenarios of requirement analysis and architecture design	Products of requirement analysis and architecture design
	Initial sizing and conceptual sketch		MOF of initial sizing and conceptual sketch		Views of initial sizing and conceptual sketch	Scenarios of initial sizing and conceptual sketch	Products of initial sizing and conceptual sketch
	Propulsion		MOF of propulsion		Views of propulsion	Scenarios of propulsion	Products of propulsion
	Configuration layout		MOF of configuration layout		Views of configuration layout	Scenarios of configuration layout	Products of configuration layout
	Overall arrangement		MOF of overall arrangement		Views of overall arrangement	Scenarios of overall arrangement	Products of overall arrangement
	Aerodynamics		MOF of aerodynamics		Views of aerodynamics	Scenarios of aerodynamics	Products of aerodynamics
	Weight		MOF of weight		Views of weight	Scenarios of weight	Products of weight
	Performance		MOF of performance		Views of performance	Scenarios of performance	Products of performance
	Cost analysis		MOF of cost analysis		Views of cost analysis	Scenarios of cost analysis	Products of cost analysis

**Fig. 2** Data transfer among disciplines.

3.3. ACDS process

According to Section 2.5, the process of aircraft conceptual design schema construction includes: acquiring data, conceptualizing, normalizing, integrating, evaluating, implementing, documenting, and maintaining. The configuration layout has a connecting link between the preceding and the following of the aircraft conceptual design and is based on the initial sizing

and conceptual sketches to form the three views, wings, tail, fuselage, landing system, nacelles, and other parameters of the aircraft design. This paper constructs a configuration layout schema as an example.

3.3.1. Acquiring data

The object of data acquisition is to obtain the model and data that can express the configuration layout of this type of air-

craft from the original data. The original data of the configuration layout are CAD models of the aircraft layout, a three-view diagram of the aircraft, “Aircraft Overall Layout Design Report”, and the wing shape data, horizontal tail shape data, vertical tail shape data, fuselage shape data, wing type data, overall parameters of the landing system, and data of engine nacelles that constitute the report document.

For the above data, through informal interviews with experts, the configuration layout instance model was formed, containing the layout description, three-view diagram, layout 3D model, wing, fuselage, horizontal tail, vertical tail, landing system, nacelle, etc. Fig. 3 shows the structure tree of the configuration layout instance model. The hierarchical structure of the configuration layout instance model is shown in the form of a structure tree, and the leaf nodes in the structure tree represent a model or data for expressing the characteristics of the aircraft configuration layout at that leaf node.

3.3.2. Conceptualizing

The instance model expresses a model of a solution for aircraft configuration layout. It is necessary to conceptualize the instance model, remove the parameters and features related to the solution, and clarify the expression of each model and data.

In the acquisition, we extract the data and documents related to the configuration layout to form the instance model and structure tree, with the leaf nodes expressed by specific model or data. The instance model conceptualization aims to clarify which model or data is used by the leaf nodes and remove the feature information related to the actual solution in the model and data. The conceptualization focuses on the type of model and data, the expression effect, and the format. For example, the instance model of span is “span = 60 m” in a certain solution. When removing the feature information, which is “60”, related to the actual solution, we can get the conceptual model. The conceptual model is that “span is a numerical type with three decimal places reserved and its unit is meter”.

The instance model of the configuration layout can be conceptualized as text, numerical, table, 3D model, and figure. Table 3 shows the representation of each instance model after the instance model conceptualization.

The instance model of the table type consists of closely related values and text. After defining rows and columns, this set of values and text constitutes an emergent effect that can express information, so it is better to express the instance model on the granularity of the table type. Table 3 fails to show the characteristics of the table type. The table type needs to define the specifics of what the rows and columns represent and in what units (Table 4).

3.3.3. Normalizing

The conceptual models of configuration layout can be classified as text, numerical, table, 3D model, and figure. Analyzing and processing the above conceptual models, extracting common features, forming specific types, and setting attributes to form a metamodel are the main work of normalization. The normalization uses the intermediate elimination method to define the metamodel first, determine the meaning and attributes of the metamodel, agree with the stakeholders on the definition of the metamodel, and then record the model specification document. Since the integrity of the metamodel cannot be proven, the idea needs to be established that new metamodels will be added at any time.

Using the intermediate elimination method, we can normalize the conceptual model into a metamodel consisting of the following elements and relationships: TEXT, NUM, TABLE, CURVE, CAD, FIG, CONTAIN relation. In terms of expressing meaning, TABLE indexes the elements through rows and columns, while CURVE is formed by discretizing the coordinates of points on a continuous multidimensional curve or surface. In terms of expression, CURVE has three characteristics: (A) only the first row contains a string, and the other rows are numbers. Furthermore, the columns of CURVE are arranged consecutively, and the cells of CURVE are not empty; (B) CURVE is generally used to draw curves in

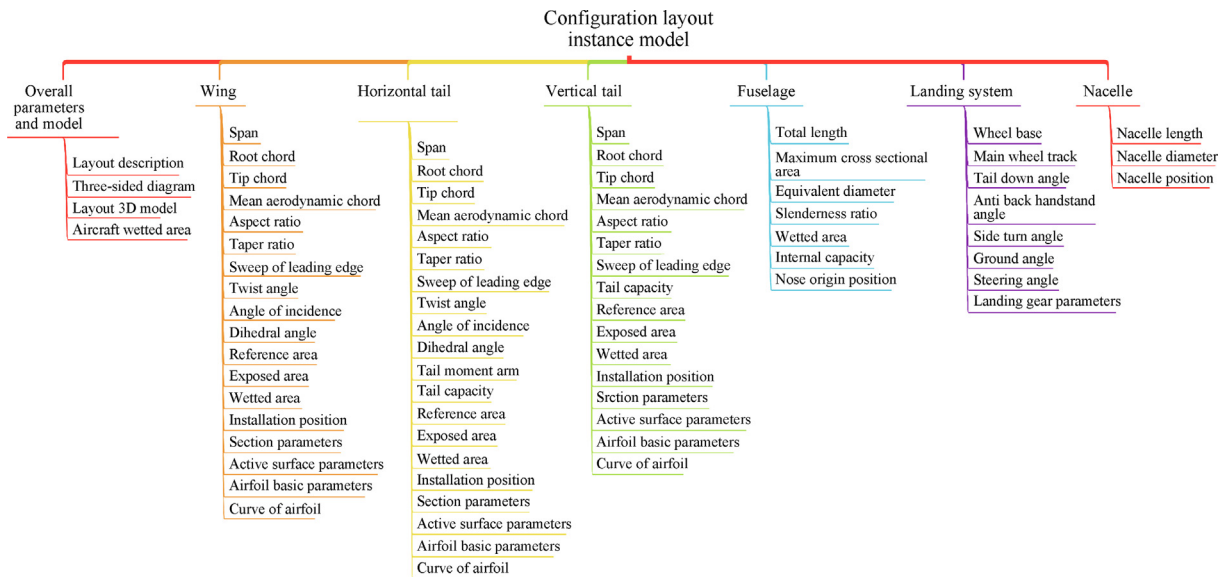


Fig. 3 Structure tree of configuration layout instance model.

Table 3 Instance model name and its conceptualizing.

Instance model name	Type	Instance model name	Type
Layout description	Text	Curve of airfoil	Table
Three-view diagram	Figure	Tail moment arm	Numerical
Layout 3D model	3D model	Tail capacity	Numerical
Aircraft wetted area	Numerical	Total length	Numerical
Span	Numerical	Maximum cross-sectional area	Numerical
Root chord	Numerical	Equivalent diameter	Numerical
Tip chord	Numerical	Slenderness ratio	Numerical
Mean aerodynamic chord	Numerical	Internal capacity	Numerical
Aspect ratio	Numerical	Nose origin position	Table
Taper ratio	Numerical	Wheel base	Numerical
Sweep of leading edge	Numerical	Main wheel track	Numerical
Twist angle	Numerical	Tail down angle	Numerical
Angle of incidence	Numerical	Anti-back handstand angle	Numerical
Dihedral angle	Numerical	Side turn angle	Numerical
Reference area	Numerical	Ground angle	Numerical
Exposed area	Numerical	Steering angle	Numerical
Wetted area	Numerical	Landing gear parameters	Table
Installation position	Table	Nacelle length	Numerical
Section parameters	Table	Nacelle diameter	Numerical
Active surface parameters	Table	Nacelle position	Table
Airfoil basic parameters	Table		

Table 4 Table type in conceptual model.

Instance model name	Row definition	Column definition
Installation position	Coordinates of the leading edge point of root	$X/Y/Z$ coordinate values
Section parameters	Different positions of a wing section	Each wing segment has the following parameters: Area of segment, half-span, root airfoil, slightly airfoil, root chord length, tip chord length, mean aerodynamic chord length, aspect ratio, sweep of leading edge, taper ratio, twist angle, incidence angle, dihedral angle.
Active surface parameters	Active surface type	Active surface area, relative reference wing area, active surface span, chord length, relative chord length, deflection angle, relative reference wing root chord
Airfoil basic parameters	Airfoil name	Airfoil leading edge radius, thickness, curvature, maximum thickness position, zero lift angle, slope of lift line, maximum lift coefficient
Curve of airfoil	A combination of a pair of relative coordinate and airfoil data	The first column shows the transverse relative coordinates of the airfoil, and the second column shows the airfoil data corresponding to the transverse coordinates.
Nose origin position	Nose origin position coordinates	$X/Y/Z$ coordinate values
Landing gear parameters	Landing gear name	Number of wheels, tire diameter, tire width, mounting position $X/Y/Z$ coordinate values, retracted form
Nacelle position	Nacelle coordinates	$X/Y/Z$ coordinate values

vertical columns in the aircraft conceptual design to express the flight time data, and part of the first column of CURVE is time;(C) the number of rows of CURVE is generally more substantial. Each cell of TABLE can include string information. Rows and columns have unique meanings. Some TABLE exists to merge cells, and the number of rows is generally less. According to the enduring implementation strategy, the first row of the curve and the rows and columns of the TABLE can be expanded on the original definition. Table 5 shows the configuration layout metamodel and its connotations and properties.

3.3.4. Integrating

The integration aims to unify duplicate models and carry out entity disambiguation and coreference resolution. In the conceptual model, the names of parameters such as span, root chord, tip chord, mean aerodynamic chord, aspect ratio, taper ratio, the sweep of leading edge, twist angle, and angle of incidence represent different meanings in the wing, horizontal tail, and vertical tail, which is an entity disambiguation problem. In general, there exist two solutions: (A) Distinguish by name. Add the specific meaning expressed in the above terms, such as wing span length and wing reference area. (B) Judging from

Table 5 Configuration layout metamodel and its connotations and properties.

Metamodel	Connotation	Property
TEXT	Configuration layout data and models that need only text to be expressed	Comments
NUM	Configuration layout parameters that need only numerical values to be expressed	Comments, units
TABLE	A set of numbers and text with a specific relationship to express the overall layout parameters	Comments, cell annotations, row definitions, column definitions, units
CURVE	A set of numbers with a specific relationship, usually plotted in columns, expressing curves	Comments, column definition, units
CAD	A model formed by CAD software, expressing the general layout model	Comments
FIG	A model expressed by a picture, expressing a three-view drawing in picture format	Comments
CONTAIN relation	Representation of the containment and inclusion relationship between two models	Parent and contained metamodel

the context, such parameters related to the wing context can be recognized as wing parameters, the metamodel “CLASS” can be defined, and “CLASS” does not have a specific meaning and can contain other models or parameters belonging to the wing classification, which can be identified as wing parameters. The second solution is adopted in the configuration layout schema.

The airfoil parameters appear equally in the wing, horizontal tail, and vertical tail and are expressed similarly for the same purpose. This coreference resolution problem is handled by stripping the airfoil parameters from them and forming a separate model.

After adjustment, the conceptual model of configuration layout contains a layout description, three-view diagram, layout model, wing, fuselage, horizontal tail, vertical tail, landing system, nacelle, and airfoil. Fig. 4 shows the conceptual model. The red area is left blank to fill in the actual scheme data to form the instance model.

Integrating should also accomplish the work of integrating metamodels created by different disciplines. Different disciplines will produce different metamodels due to different modeling purposes and modeling tools and different engineers' perceptions. Some of the metamodels may have the same characteristics but different names.

3.3.5. Evaluating

With the help of discipline engineers, evaluation will consist of two processes: verification and validation. The specific behavior of the validation process is presented as follows. The conceptual model and instance model of configuration layout containing instance data are sent down to the overall design engineer. The overall design engineer verifies the accuracy of the instance model representation, and then uses the conceptual model to generate another instance model to verify the usability. The specific actions of the validation are that the overall design engineer confirms that the conceptual model is available and that downstream discipline engineers such as the overall engineer confirm that the conceptual model contains their required model.

The evaluation also includes the evaluation of ASoT and design aids. The coordination of the schema, the ASoT, and design aids are evaluated. The ease of use of the ASoT and design aids is assessed, and modifications are proposed and iteratively tested.

3.3.6. Implementing

After evaluation and iteration by engineers, the trial was extended. Also, we distribute the constructed ASoT along with design aids to try them out in projects.

3.3.7. Documenting

The configuration layout schema was created to form data source, conceptualization, specification, integration, evaluation, implementation, and maintenance documents. Also, the schema usage instructions document is formed.

3.3.8. Maintaining

The Schema needs to be maintained regularly, updated, and modified appropriately. According to the enduring implementation strategy, the design solution is constantly changing, and some of the solutions may result in new aircraft configurations. For example, if the new configuration of the aircraft contains a V-tail and does not contain a horizontal tail and a vertical tail, the configuration layout schema should be modified accordingly for compatibility.

3.4. ACDS MOF

After acquiring data, conceptualizing, and normalizing, we get the instance model, conceptual model, and metamodel of the aircraft conceptual design, and the metamodel of schema is “element” and “relationship” as required by the methodology. The metamodel is the instantiation of the meta metamodel, the elements are TEXT, NUM, TABLE, CURVE, CAD, FIG, REQ, SYS, CAE, ANNEX, and CLASS, and the relationships are CONTAIN, REF, TRACE, and SATISFY relations. Table 6 shows the connotation and properties of elements in the metamodel. Table 7 shows the connotation and related elements of relationships in the metamodel.

The conceptual model is an instantiation of the metamodel, combined with the metamodel and given specific properties. The conceptual model for each discipline of aircraft conceptual design is defined, and the metamodels contained in the conceptual model and the metamodel uses are described. CLASS is widely used in conceptual models as a universal hierarchical organization metamodel.

NAME	PARENT	TYPE	VALUE	UNIT	COMMENT	NAME	PARENT	TYPE	VALUE	UNIT	COMMENT
Layout description		TEXT				Vertical tail		CLASS			
Three view diagram		FIG				Span	Vertical tail	NUM		m	
Layout 3D model		CAD				Root chord	Vertical tail	NUM		m	
Aircraft wetted area		NUM		m ²		Tip chord	Vertical tail	NUM		m	
Wing		CLASS				Mean aerodynamic chord	Vertical tail	NUM		m	
Span	Wing	NUM		m		Aspect ratio	Vertical tail	NUM			
Root chord	Wing	NUM		m		Taper ratio	Vertical tail	NUM			
Tip chord	Wing	NUM		m		Sweep of leading edge	Vertical tail	NUM		*	
Mean aerodynamic chord	Wing	NUM		m		Tail capacity	Vertical tail	NUM			
Aspect ratio	Wing	NUM				Reference area	Vertical tail	NUM		m ²	
Taper ratio	Wing	NUM				Exposed area	Vertical tail	NUM		m ²	
Sweep of leading edge	Wing	NUM		*		Wetted area	Vertical tail	NUM		m ²	
Twist angle	Wing	NUM		*		Installation position	Vertical tail	TABLE			
Angle of incidence	Wing	NUM		*		Section parameters	Vertical tail	TABLE			
Dihedral angle	Wing	NUM		*		Active surface Parameters	Vertical tail	TABLE			
Reference area	Wing	NUM		m ²		Airfoil		CLASS			
Exposed area	Wing	NUM		m ²		Airfoil basic parameters	Airfoil	TABLE			
Wetted area	Wing	NUM		m ²		Curve of airfoil	Airfoil	CURVE			
Installation position	Wing	TABLE									
Section parameters	Wing	TABLE									
Active surface parameters	Wing	TABLE									
Horizontal tail		CLASS									
Span	Horizontal tail	NUM		m		Fuselage		CLASS			
Root chord	Horizontal tail	NUM		m		Total length	Fuselage	NUM		m	
Tip chord	Horizontal tail	NUM		m		Maximum cross sectional area	Fuselage	NUM		m ²	
Mean Aerodynamic chord	Horizontal tail	NUM		m		Equivalent diameter	Fuselage	NUM		m	
Aspect ratio	Horizontal tail	NUM				Slenderness ratio	Fuselage	NUM			
Taper ratio	Horizontal tail	NUM				Wetted area of fuselage	Fuselage	NUM		m ²	
Sweep of Leading edge	Horizontal tail	NUM		*		Internal capacity	Fuselage	NUM			
Twist angle	Horizontal tail	NUM		*		Nose origin Position	Fuselage	TABLE			
Angle of incidence	Horizontal tail	NUM		*		Landing gear parameters		CLASS			
Dihedral angle	Horizontal tail	NUM		*		Wheel base	lg Gear Para	NUM		m	
Tail moment arm	Horizontal tail	NUM		m		Main wheel track	lg Gear Para	NUM		m	
Tail capacity	Horizontal tail	NUM				Tail down angle	lg Gear Para	NUM		*	
Reference area	Horizontal tail	NUM		m ²		Anti-Back handstand angle	lg Gear Para	NUM		*	
Exposed area	Horizontal tail	NUM		m ²		Side turn angle	lg Gear Para	NUM		*	
Wetted area	Horizontal tail	NUM		m ²		Ground angle	lg Gear Para	NUM		*	
Installation position	Horizontal tail	TABLE				Steer angle	lg Gear Para	NUM		*	
Section parameters	Horizontal tail	TABLE				Landing Gear Parameters	lg Gear Para	TABLE			
Active surface parameters	Horizontal tail	TABLE				Nacelle		CLASS			
						Nacelle length	Nacelle	NUM		m	
						Nacelle diameter	Nacelle	NUM		m	
						Nacelle position	Nacelle	TABLE			
Airfoil	Leading edge radius	Thickness(%)	Curvature(%)	Maximum thickness position(%)	Zero lift angle(°)	Lift curve slope	Maximum lift coefficient				
NACA 4415											
NACA 0012											
NACA 1412											
N 60R											
Location Name	X(mm)	Y(mm)	Z(mm)								
Engine nacelle coordinates											
Number of wheels	Tire diameter(mm)	Tire width(mm)	Mounting position X(mm)	Mounting position Y(mm)	Mounting position Z(mm)	Storage form					
Front landing gear											
Main landing gear 1											
Main landing gear 2											

Fig. 4 Conceptual model of aircraft configuration layout.

- (1) Requirements Analysis and Architecture Design contains REQ and SYS. REQ is used to express the aircraft design requirements; SYS is used to express the aircraft functional architecture SysML model.
- (2) Conceptual Sketch and Initial Sizing contains TEXT, NUM, CURVE, FIG, and ANNEX. TEXT is used for concept sketch description. NUM is used to express overall parameters such as wing load, thrust-to-weight ratio, and other parameters, and the parameters are given unit attributes. CURVE is used to express the constraint boundary curve, the first column of the constraint boundary curve is wing load, and the rest columns are constraint values of wing load corresponding to thrust-to-weight ratio. FIG is used to present conceptual sketches, and ANNEX is used for the detailed description of the conceptual sketches.
- (3) Propulsion includes TEXT, NUM, FIG, CAD, CURVE, and ANNEX. TEXT is used to express the engine type and basic description. NUM describes the dimensional engine parameters, weight, and basic performance parameters such as the culvert ratio and static thrust. FIG is used to display multiple engine pictures. CAD is used for the engine 3D model. ANNEX is used for the engine presentation documents; CURVE is used for the engine thrust and fuel consumption rate curves. Both curves contain the engine maximum, cruise, and jog states, with Ma in the first row and thrust or fuel

consumption rate at a given altitude in the remaining rows. Units are defined for both the numerical and curve columns.

- (4) Configuration layout includes TEXT, NUM, FIG, TABLE, CURVE, and CAD. TEXT is used for layout description. FIG is used for a three-view diagram. CAD is used for the overall layout 3D model. NUM and TABLE are used for overall layout parameters, wing parameters, horizontal tail parameters, vertical tail parameters, fuselage parameters, etc. CURVE is used to represent airfoil data. NUM, TABLE, and CURVE are defined as units.
- (5) Overall Arrangement includes TEXT, NUM, FIG, TABLE, and CAD. TEXT defines the layout description and the form of the passenger cabin and cargo cabin layout; NUM describes the basic dimensions of the passenger cabin and cargo cabin; FIG shows the overall arrangement diagram; CAD represents the overall layout 3D model; TABLE represents the passenger cabin layout parameters.
- (6) Aerodynamics analysis includes NUM, CURVE, and CAE. NUM defines aerodynamic parameters such as maximum lift coefficient; CURVE defines aerodynamic analysis curves such as lift characteristics, drag characteristics, moment characteristics, and other curves; CAE defines CFD calculation results. All the aerodynamic parameters are dimensionless. The first column

Table 6 Elements and their connotation and properties.

Element	Connotation	Property
TEXT	Model information that can be expressed as a string only	Comments
NUM	Model parameters that can be expressed as floating point numbers or integers only	Comments, units
TABLE	Tables, a set of data and strings in tabular form, with specific relationships between rows and columns	Comments, cell units, cell annotations, row definitions, column definitions
CURVE	In CURVE data, only the first row contains a string, multiple columns of floating point numbers or integers, with a certain relationship between the columns	Comments, column units, column definition
CAD	A model formed by CAD software that can express 3D information	Comments
FIG	Visual representation of the model in common image format	Comments
REQ	Requirements model, which defines the aircraft design requirements and contains documents formed by various requirements management tools and requirements standardization files in Reqif format	Comments, attribute column definition
SYS	System model, a SysML model that defines the functional architecture of the aircraft and can be formed by various MBSE tools	Comments
CAE	CAE model formed by CAE software, such as CFD, FEM, and other tools	Comments
ANNEX	Attachments, which do not need to parse the content of the file but need to form a file for data transfer or archiving, and support any type, and some types can be extended into separate element types	Comments
CLASS	Classification, to help achieve hierarchical organization of relationships	Comments, parent element, child element

Table 7 Relationships and their connotation and related elements.

Relationship	Connotation	Related element
CONTAIN	Containing relationships, which define those one or more elements contain another element in order to build hierarchical relationships	Parent element, child element
REF	Reference relationship, which defines that a model needs to depend on one or more specific elements of the model to carry out the design	Referenced elements
TRACE	Traceability relationships, which define one or more arbitrary elements of a model that need to be traced back to a requirement model, are assertions that the arbitrary element will satisfy the requirement	Arbitrary element, traced requirement model
SATISFY	Satisfaction relationship, where one or more arbitrary elements are defined to satisfy a requirement model, requires a determination of whether the requirement is met	Arbitrary element, satisfied requirement model

of the aerodynamics analysis curve is the angle of orientation, and the rest are the aerodynamics parameters of different Ma.

- (7) Weight analysis includes TABLE, which is used for the weight calculation table, the center of gravity front and rear limit table, takeoff and landing weight calculation table, and other tables, and the unit of weight is kg.
- (8) Performance analysis includes NUM, CURVE, and TABLE. NUM is used to express single performance parameters such as takeoff equilibrium field length, theoretical static lift limit, etc. CURVE describes the performance curves of takeoff, climb, cruise, circling, descent, landing, etc. Each curve defines the first row and unit. TABLE is used to represent the mission and route.
- (9) Cost analysis includes NUM for indicating the economy index and giving units.

The conceptual model of ACDS does not contain specific parameters or curve values, and the instance model is formed by filling in the data and model. The schema constructed in this paper is closely related to the ASoT and its design aids. The instance model can be submitted through custom develop-

ment to the ASoT, and the design results can be managed through the ASoT.

3.5. ACDS views

The main stakeholders in the conceptual design phase are the project leader, team leader, and conceptual design engineers, among which the conceptual design engineers include requirements engineer, system engineer, overall design engineer, propulsion engineer, aerodynamics engineer, weight engineer, performance engineer, and cost analysis engineer.

According to stakeholders' deconstruction of the conceptual design phase, the aircraft conceptual design view set can be sorted out (Table 8). The view sets can be divided into two categories: interdisciplinary views and discipline-specific views.

3.6. ACDS key scenarios

Aircraft conceptual design key scenarios are closely linked to the design phase, with each discipline having its specific design scenario and involving related views. There are also interdisci-

plinary design scenarios in aircraft design that require collaboration between disciplines (Table 9).

3.7. ACDS products

According to Section 2.8, the following products should be included:

- (1) An ACDS covers the standardized representation of the aircraft conceptual design phase of requirement analysis and architecture design, initial sizing and conceptual sketch, configuration layout, overall arrangement, propulsion, aerodynamics, weight, performance, and cost analysis. The instance model, conceptual model, and metamodel are included. The ACDS can be accessed through “<https://gitee.com/sun-xiaojian/ACDS>”.

- (2) A set of documents. During the schema construction process, data source documents, conceptualization documents, specification documents, integration documents, evaluation documents, implementation documents, and maintenance documents are formed. Also, a schema usage instruction document is formed.
- (3) An ASoT and a set of design aids.

4. Discussion of schema features

4.1. Support for enduring aircraft design

Aircraft design begins with conceptual ideas and progresses through conceptual design to preliminary design and then to detailed design, where the number of participants grows and

Table 8 ACDS views.

View type	View	Connotation
Interdisciplinary views	Aircraft design parameters set	The project leader is concerned with the critical design data of the aircraft, which is extracted from the data of various disciplines.
	Propulsion selection input view	The parameters of concern to the propulsion engineer require input data obtained from the disciplines of requirements analysis, architecture design, initial sizing, etc., through which the data combined with tools can carry out the work of propulsion data analysis.
	Overall arrangement input view	The data of interest to the overall arrangement engineer requires input data obtained from the disciplines of requirements analysis, architecture design, and propulsion, as well as conceptual sketches and initial sizing, configuration layout, etc., through which the design work of the overall arrangement can be carried out.
	Aerodynamics analysis input view	The input data of concern to the aerodynamics engineer includes the data of aerodynamics design requirements and the geometric model of the whole machine obtained from the requirements analysis, configuration layout, and other specialties.
	Weight calculation input view	The input data for weight calculation engineers includes the aircraft's weight index and layout data obtained from requirements analysis, configuration layout, and other disciplines.
	Performance analysis input view	The input data of interest to the performance analysis engineer includes the aerodynamic data, weight center of gravity data, and propulsion data obtained from aerodynamic, weight, and propulsion disciplines.
	Cost analysis input view	Input data of interest to cost analysis engineers, including data obtained from requirements analysis, performance, and other disciplines.
Discipline-specific views	Requirements analysis and architecture design view	The data that the requirements team leader is concerned about, including the data of all the deliverables from the requirements engineers, represents the work of the requirements team.
	Initial sizing and conceptual sketch view	Overall team leader focuses on the overall parameters, including data from initial sizing, aircraft conceptual sketches, etc.
	Propulsion selection view	The propulsion team leader's data includes propulsion engineers' deliverables data, such as the basic engine parameters after propulsion selection and engine performance data.
	Configuration layout view	The overall team leader is concerned with the layout data, including the aircraft's basic configuration and shape data.
	Overall arrangement view	The overall team leader focuses on the layout data, including the aircraft's passenger cabin, cargo cabin, cockpit layout, and primary system layout data.
	Aerodynamics analysis view	The aerodynamics team leader focuses on the data, including the aerodynamic number table in various states, reflecting the results of the work of the aerodynamics design engineers.
	Weight calculation view	The weight team leader focuses on the data, including the aircraft's weight center of gravity, inertia, and other data.
	Performance analysis view	Performance team leaders focus on the data, including the aircraft's takeoff and landing, cruise, hover, and other performance data.
	Cost analysis view	The data in charge of cost analysis includes the results of the work of economic analysis engineers: Direct Operating Cost (DOC), Total Operating Cost (TOC), break-even frame count, and other indicators.

Table 9 ACDS key scenarios.

Scenario	Input	Output	Stakeholder	Connotation
Solution trade-off analysis	Aircraft design parameters set	Preferred solution	Project leaders	Use aircraft design parameter sets from multiple options, compare and analyze them, and weigh them to arrive at the better option.
Scenarios of requirements analysis and architecture design	Stakeholder needs	Requirements analysis and architecture design view	Requirements team leader, requirements engineer, system engineer	Use stakeholder needs to form aircraft-level requirements and analyze and refine the requirements to perform targeted architectural design to form aircraft subsystem-level requirements.
Scenarios of initial sizing and conceptual sketch	Requirements analysis and architecture design view	Initial sizing and conceptual sketch view	Overall team leader, overall design engineer	The requirements analysis and architecture design results are used to carry out the activities of overall parameter selection and design and conceptual sketch design.
Scenarios of propulsion	Propulsion selection input view	Propulsion selection view	Propulsion team leader, propulsion engineer	Based on the initial sizing, carry out propulsion selection and evaluation by the requirements for propulsion in the requirements analysis.
Scenarios of configuration layout	Initial sizing and conceptual sketch view	Configuration layout view	Overall team leader, overall design engineer	Using the conceptual sketch as a blueprint and the overall parameters, model the overall layout and determine the layout parameters.
Scenarios of overall arrangement	Overall arrangement input view	Overall arrangement view	Overall team leader, overall design engineer	Based on the propulsion analysis results and the configuration layout's shape, build the aircraft's internal layout model and determine the internal layout parameters.
Scenarios of aerodynamics	Aerodynamics analysis input view	Aerodynamics analysis view	Aerodynamics team leader, aerodynamics engineer	Based on the shape of the configuration layout, carry out aerodynamic calculations and revise the configuration layout shape model and parameters if necessary.
Scenarios of weight	Weight calculation input view	Weight calculation view	Weight team leader, weight calculation engineer	Calculate the weight center of gravity of the aircraft from the weight and center of gravity of each component in the overall arrangement, and carry out an analysis compared with the weight requirements.
Scenarios of performance	Performance analysis input view	Performance analysis view	Performance team leader, performance analysis engineer	Use complete aircraft design parameter sets from multiple options, compare and analyze them, and weigh them to arrive at the better option.
Scenarios of cost analysis	Cost analysis input view	Cost analysis view	Cost team leader, cost analysis engineer	Use stakeholder needs to form aircraft-level requirements and analyze and refine the requirements to perform targeted architectural design to form aircraft subsystem-level requirements.

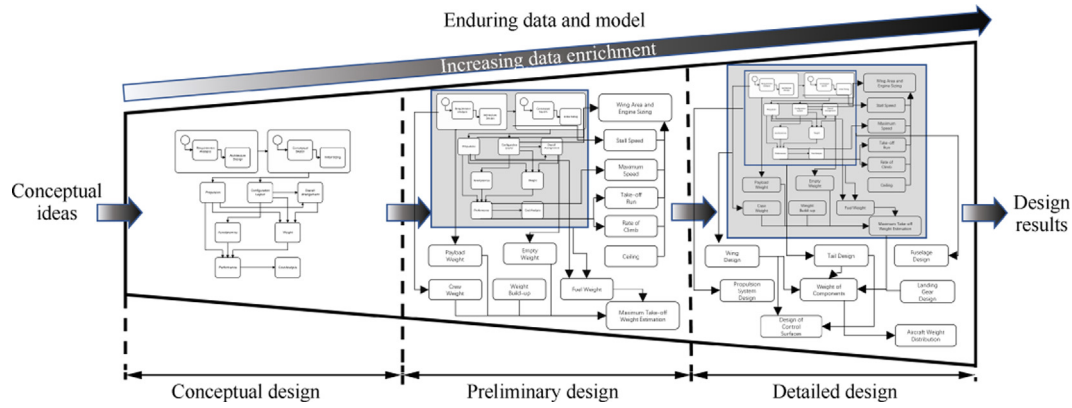


Fig. 5 Increasing data enrichment in aircraft design.

the solutions become complex. The granularity of models and data refines the quantity and variety of changes and increases data enrichment. However, enduring data and models' essential elements and relationships are enumerable. As shown in Fig. 5, models and data can be composed of different combinations of elements and relationships. As the design progresses, the discipline gradually becomes more and the enrichment of the data becomes higher.

As required by the methodology, new metamodels will be added at any time, and the schema provides an open-ended structure where the kinds of elements and relationships can be openly expanded. Newly expanded metamodels do not affect old ones, which can be modified incrementally.

This approach can support enduring aircraft design. During the transition of decision gates between phases, the representation of models within a discipline often changes considerably, and an open schema can support changes in the way metamodels are represented, keeping old models, and adding new ones. In the case of tables, the complexity of the data generated by an enduring detailed design increases, and the rows and columns of the original tables are no longer sufficient to represent the data and models. It is common to encounter very different models when exchanging models among different disciplines at the same stage. One of the principles is to consider data transfer between disciplines, so schema construction according to the methodology requires

little or no more work to accommodate model transfer between disciplines naturally.

4.2. Support for digital engineering ecosystem

4.2.1. Preliminary form of digital engineering ecosystem

The preliminary form of the digital engineering ecosystem consists of ACDS, the ASoT and design aids (Fig. 6). ACDS is the digital base, which helps engineers normalize their models and data. The ASoT is used to manage the data and models, which conform to the ACDS standard format. Models and data generated during the aircraft design process need to be uploaded and approved to the ASoT, which is also the source of models and data for design and analysis during the design process. Design aids include both software and hardware environments. The software environments include in-house tools and commercial software and its secondary development. The software is capable of reading and generating files in ACDS format to help engineers reduce the design burden of forming delivery data and reports during the design process. Hardware includes flight simulators and XR devices. Based on ACDS format files, the hardware is supported to quickly access the design solution to start simulation analysis. The preliminary form of the digital engineering ecosystem is not yet complete, and needs to be gradually formed into a complete digital

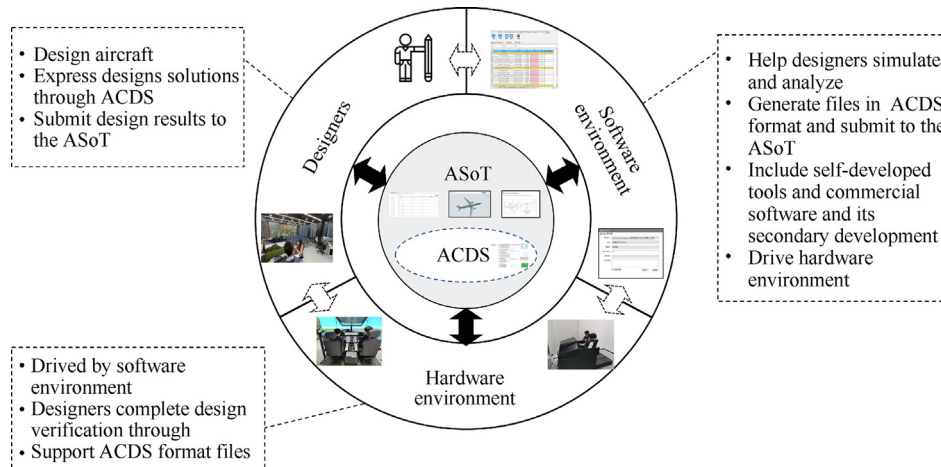
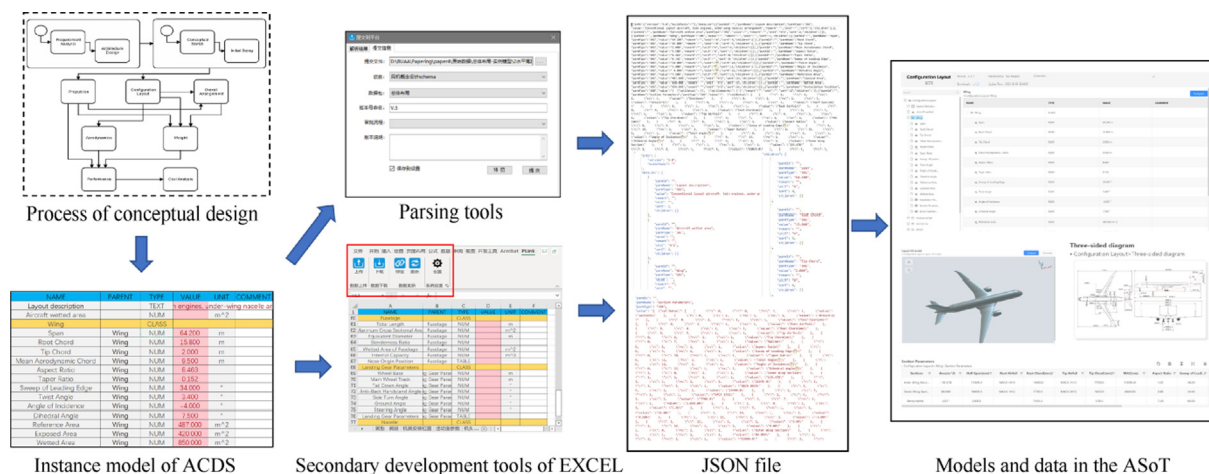


Fig. 6 Preliminary form of digital engineering ecosystem.

**Fig. 7** Parsing and uploading instance model to ASoT.

engineering ecosystem by virtue of the continuous introduction of new technologies and the propagation of concepts.

4.2.2. Reducing the design burden with design aids

One of the principles is to focus on the data generated by the design aids. The instance model is part of the input to those tools. Design aids consist of document-specific parsing tools and software-specific secondary development tools. The schema supports the secondary tools with a four-layer MOF. The parsing tool is concerned with the parsing and refactoring elements and relationships, not with the conceptual and instance models. Parsing tools can parse files in specified directories and help engineers upload instance models from local paths to the ASoT. Secondary development tools are embedded in specific industrial software, providing input conditions and forming instance models. Fig. 7 shows design aids and the parsing results. By defining a specific parsing form for each element and relationship, design aids convert the schema into a computer-readable JSON file and generate a counterpart in the ASoT, which allows the instance model to be quickly uploaded to the ASoT.

4.2.3. Support for dynamic management of ASoT

The schema supports a variety of models and data management methods for the ASoT. The dynamic management of the ASoT includes version management, relationship management, scheme management, and view management. The schema's multiple relational metamodels can be explicitly represented in the ASoT, and relationships can be changed at any time to enable dynamic relationship management. The schema's fine-grained data and models can be combined to form new views, and the schema defines a set of views that can be dynamically managed through the ASoT. Fig. 8 shows an example of a schema's configuration layout model expressed in the ASoT, which contain version management, relationship management, scheme management, and view management.

4.3. Features of ACDS

There are several pain points in aircraft design. One of the pain points is the trade-off among multiple solutions during the

conceptual design phase. There are many different design solutions for an aircraft. Due to many solutions, the reuse process is prone to misuse, and a wrong parameter reference may lead to the failure of a solution. The complex referencing relationship between the disciplines within a solution, and even the phenomenon of circular referencing, make the boundaries of the disciplines unclear and prevent engineers from clarifying their responsibilities. On the other hand, model-based design is not completely widespread, and current design solutions are still presented in documented reports, with the solution data implicitly included in the reports. The trade-offs between solutions are based on multiple reports and cannot be directly compared based on the model's convenience, which multiplies the burden on decision makers.

Multiple iterations are another design pain point. Conceptual solutions need to go through multiple iterations in a short cycle, with each solution requiring calculations and reports, leaving engineers overwhelmed and under pressure. In the absence of technical status management, model reuse is prone to versioning errors, data inconsistencies, and in severe cases, rework. As iterations occur among many engineers and result in a small number of deliverables and reports, a large amount of work and models are not visible to project managers, so the progress of multiple iterations cannot be monitored uniformly, and the status of iterations cannot be updated in real time.

Multidisciplinary coupling is also one of the pain points in aircraft design, which involves a wide range of disciplines and complex coupling relationships between design models. The interfaces between disciplines are not fixed, and often one interface is used for one solution or configuration, significantly limiting the reusability of models.

The main reason for these three pain points is the weak and coarse granularity of model and data management, making it impossible to achieve fine-grained technology state management. Each discipline organizes its models most conveniently during the design process, models and data do not adopt a uniform schema, and interfaces between disciplines are not fixed. Another reason is that the association and referencing of models cannot be established.

The schema implementation strategy is that using schema can reduce engineers' workload. ACDS can uniform terminology names and definitions. Communication among different

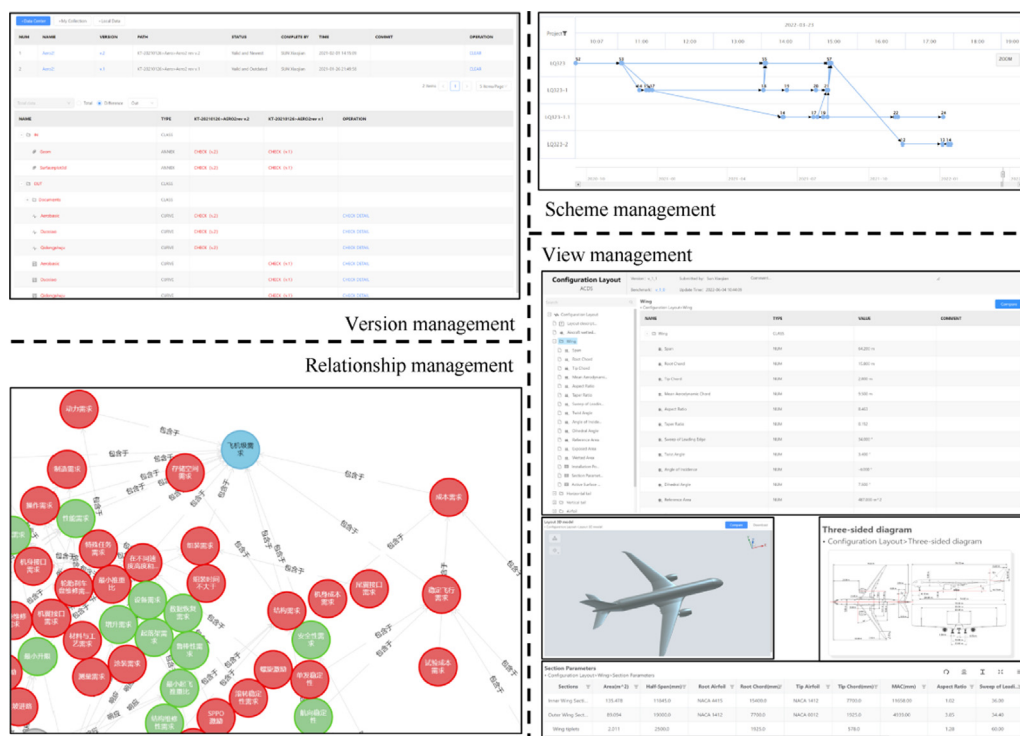


Fig. 8 Configuration layout model expressed in ASoT.

teams is based on the same set of terms (ACDS), so ACDS can reduce communication time. After the data is packaged, parsed, and uploaded by the tool to the ASoT, automatic report generation can be performed. Report writing for engineers used to take days, but now it takes minutes. All the data of the design process can be retrieved directly in the ASoT, without the need to spend a lot of time looking through documents to find data. When the data of upstream discipline changes, the upstream engineer does not need to send emails to remind the downstream engineer, and the downstream engineer also does not need to ask the upstream engineer for the data, ASoT can automatically remind and manage the data.

By creating a schema, designers can accurately trade off multiple design solutions, quickly iterate through multiple rounds of design and sort out multidisciplinary coupling.

(1) Helping designers design well

The essence of schema is to transform the current document report-based collaborative design into a digital model and data-based collaborative design paradigm. ACDS breaks up the document report into models and data such as values, text, tables, curves, and 3D models, refining the granularity of model and data management. Multiple solutions are expressed in the same conceptual schema, making the instance models implicit in the documentation report visible. With the help of the ASoT and design aids, the trade-offs among multiple solutions can be quickly analysed, and the advantages and disadvantages of critical parameters can be derived. Managing fine-grained models and data makes data references between scenarios explicit, allowing data references to be traced back to the source, clarifying professional boundaries and responsibilities, and breaking the circularity of references.

(2) Helping designers design fast

ACDS reduces the time spent in compiling documentation by creating fine-grained models and data based on uniform documentation templates that can automatically generate reports. The results of the simulation tools are generated directly into instance models via design aids, reducing the time required to organize the instance models. The instance models of design solutions can be uploaded to the ASoT in real time with design aids, enabling version management of data models, professional deliverables, and iterative solutions for the whole machine. By uploading engineers' data and models in real time, project management can monitor the progress of iterations and the status of solutions in a unified manner. The consistency and reusability of fine-grained models and data under unified management will be enhanced, and model reuse between solutions will be more convenient.

(3) Helping designers design right

ACDS establishes fixed interfaces between professions and is based on the concept that discipline interfaces can be expanded without affecting old interfaces, clarifying the interfaces among multiple scenarios and configurations and increasing the reusability of the model.

ACDS's metamodel contains a variety of relationships. The model and data can support the construction of reference, traceability, and satisfaction relations. Reference relations help clarify the complex coupling between parameters, while traceability and satisfaction relations help verify and validate requirements and ensure that designers form the right solution.

5. Conclusions

- (1) In order to improve the efficiency of aircraft collaborative design information sharing and to solve the pain points of multiple solution trade-offs, multiple iterations, and multidisciplinary coupling that occur in aircraft collaborative design, this paper argues that an aircraft design schema needs to be constructed for managing fine-grained models and unifying the representation of models. This paper creatively proposes a methodology for constructing an aircraft design schema. The methodology includes implementation strategies, construction principles, processes, meta-object facility, views, scenarios, products, and a construction framework representing a Zachman style matrix.
- (2) To help users understand and apply the methodology, this paper focuses on the aircraft conceptual design phase and forms the ACDS by the methodology. The aircraft conceptual design phase is an SoI for the methodology and contains many disciplines, such as requirement analysis and architecture design, initial sizing and conceptual sketch, configuration layout, overall arrangement, propulsion, aerodynamics, weight, performance, and cost analysis. ACDS contains metamodels, conceptual models, and instance models for each discipline and forms a view set, scenario set, and product set for the aircraft conceptual design phase.
- (3) This paper specifies the features of the aircraft design schema and analyses the schema's support for coherent revolutionary aircraft design, the authoritative source of truth, and collaborative design pain point issues.

Aircraft collaborative design requires the involvement of many people and disciplines. ACDS, the data and modeling standard for aircraft design, unifies the terminology used by engineers. ACDS specifies the input and output standards for each discipline and can facilitate the flow of information between disciplines. Engineers work collaboratively through ACDS and the ASoT. It can improve the efficiency of data flow by obtaining data that they need from the ASoT.

In conclusion, the schema construction methodology proposed in this paper is functionally complete, logically self-consistent, and implementable, and can be gradually extended after pilot application in some disciplines. The schema construction methodology can gradually promote the transition from document-based collaborative design to a model-based digital engineering ecosystem. The preliminary form of digital engineering ecosystem needs to be strengthened based on the schema.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Nagel B, Böhnke D, Gollnick V, et al. Communication in aircraft design : Can we establish a common language? *28th congress of the international council of the aeronautical sciences 2012*. Brisbane Australia. 2012. p. 1–13.
2. Bohm MR, Stone RB, Simpson TW, et al. Introduction of a data schema to support a design repository. *Comput Aided Des* 2008;**40** (7):801–11.
3. US Department of Defense. DoD digital engineering strategy. Washington, D.C.: Office of the Deputy Assistant Secretary of Defense for Systems Engineering; 2018.
4. Noguchi RA, Wheaton MJ, Martin JN. Digital engineering strategy to enable enterprise systems engineering. *INCOSE Int Symp* 2020;**30**(1):1727–41.
5. MBSE Wiki. mbse:authoritative_source_of_truth [Internet]. Milford: Object Management Group, Inc.; 2020 [cited 2022 Jul 9]. Available from: http://www.omgwiki.org/MBSE/doku.php?id=mbse:authoritative_source_of_truth. n.d.
6. Sun X, Luo M, Zhang C, et al. Construction technology of authoritative source of truth for civil aircraft pre-research. *Acta Aeronaut Astronaut Sinica* 2021;**42**(2):224222 [Chinese].
7. Zill T, Böhnke D, Nagel B. Preliminary aircraft design in a collaborative multidisciplinary design environment. *11th AIAA aviation technology, integration, and operations (ATIO) conference*; Virginia Beach, VA. Reston: AIAA; 2011.
8. Crisp HE. Systems engineering vision 2020. Seattle, VA: International Council on Systems Engineering (INCOSE); 2007.
9. Dubos GF, Coren DP, Kerzhner A, et al. Modeling of the flight system design in the early formulation of the Europa Project. *2016 IEEE aerospace conference*; 2016 Mar 5-12; Big Sky, MT, USA. Piscataway: IEEE; 2016. p. 1–14.
10. Bayer T, Chung S, Cole B, et al. Update on the model based systems engineering on the Europa mission concept study. *2013 IEEE aerospace conference*; 2013 Mar 2-9; Big Sky, MT, USA. Piscataway: IEEE; 2013. p. 1–13.
11. Bayer T, Chung S, Cole B, et al. Early formulation model-centric engineering on NASA's Europa mission concept study. *INCOSE Int Symp* 2012;**22**(1):1695–710.
12. Bayer TJ, Chung S, Cole B, et al. Model based systems engineering on the Europa mission concept study. In: *2012 IEEE aerospace conference*; 2012 Mar 3-10; Big Sky, MT, USA. Piscataway: IEEE; 2012. p. 1–18.
13. Sun X, Luo M, Cui Z, et al. A collaborative design method for civil aircraft based on an enduring and evolutionary authoritative source of truth. *33th international congress of the aeronautical sciences*; Stockholm, Sweden. 2022. p. 1–16.
14. Zachman JA. A framework for information systems architecture. *IBM Syst J* 1987;**26**(3):276–92.
15. INCOSE. *System engineering handbook: A guide for system life cycle processes and activities*. 4th ed. Hoboken: John Wiley & Sons, I.; 2015.
16. He D, Zhao Y, Guo B, et al. *COMAC systems engineering manual*. 4th ed. Shanghai: Shanghai Jiao Tong University Press; 2021.
17. Tang W. Meta object facility. In: Liu L, Tamer ÖM, editors. *Encyclopedia of database systems*. Boston, MA: Springer, US; 2009. p. 1722–3.
18. Meta object facility-CIMI [Internet]. Wikipedia. [cited 2022 May 2]. Available from: http://informatics.mayo.edu/CIMI/index.php/Meta_Object_Facility.
19. Batini C, Lenzerini M. A methodology for data schema integration in the entity relationship model. *IEEE Trans Softw Eng* 1984;**SE-10**(6):650–64.
20. Fernandez M, Gomez-Perez A, Juristo N. Methontology: From ontological art towards ontological engineering. 1997. p. 8.
21. Raymer DP. *Aircraft design: A conceptual approach*. 6th ed. Reston: AIAA; 2018.
22. Mavris DN, Pinon OJ. A systems engineering approach to aircraft design. *Encyclopedia of aerospace engineering*. Hoboken: John Wiley & Sons, Ltd; 2012.