Received October 5, 2021, accepted October 23, 2021, date of publication October 27, 2021, date of current version November 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123348

An Authoritative Source of Truth System for UCAV Conceptual Design

XIAOJIAN SUN, MINGQIANG LUO, ZHIYANG CUI, ZHICHENG ZOU, AND HU LIU
School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China

Corresponding author: Mingqiang Luo (luomingqiang_buaa@163.com)

ABSTRACT Model-Based Systems Engineering (MBSE) is a part of a long-term trend toward model-centric approaches. The Authoritative Source of Truth (AST) is a crucial step of digital transformation and a core component of MBSE. Benefits include help design teams shorten the design cycle, improve design efficiency, and ensure the accuracy of design models and data compared to traditional document-based co-design methods. Data-driven design is a new direction of Unmanned Combat Aerial Vehicles (UCAV) design, and UCAV is the main force of future air warfare. How to build a UCAV conceptual design AST system is a problem that needs to be solved. We design an AST construction methodology, develop an AST system by applying the methodology, and use the UCAV concept design case to test the usability of the AST system. More specifically, the methodology includes construction goals and objects, basic construction methods, AST architecture, collaborative construction methods, dynamic management methods, basic functions and derivative tools required. In addition, the validation process is requirement analysis, functional architecture design, scheme design, and dynamic management validation. The AST system can help the implementation of MBSE in the field of aircraft design. The AST system can also help meet the requirements of the conceptual design stage, which shows the feasibility of the construction methodology's integrity. If the AST system is used in the aircraft's detailed design phase, further models and data will be required, and the AST system will need to be upgraded.

INDEX TERMS Authoritative source of truth, model-based systems engineering, UCAV, models and data, conceptual design.

I. INTRODUCTION

Aircraft design and spacecraft design are mainly based on system engineering [1], [2]. Model-Based Systems Engineering(MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [3], [4]. NASA has successfully applied the MBSE method in several projects [5]–[7], such as the Europa project. Furthermore, MBSE has also been applied to aero-engines design [8], aircraft design [9], satellite communication system architecting [10], and an integrated system design [11] by other researchers. They hold the position that the models realized by the MBSE method are reusable and easily extendible to detailed system design and implementation in the entire product life cycle.

MBSE is a part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical, and software. It emphasizes formalized modeling, has a long training cycle, and is challenging to implement. The Authoritative Source of Truth (AST) comprises unified models and data across the MBSE process. The AST provides model management that enhances knowledge acquisition and information reuse. The AST also can maintain consistency across professional models throughout the life cycle by providing a structured and interrelated representation. In short, the AST is a crucial step towards model-centric implementation and a robust foundation for the MBSE to be implemented step by step.

The AST, a core element of MBSE implementation, is one of five goals that make up the U.S. Department of Defense's Digital Engineering Strategy [12]. This goal moves the primary means of communication from documents to digital models and data. The AST enables access, management, analysis, use, and distribution of information from a standard

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

set of digital models and data. As a result, stakeholders have the current, authoritative, and consistent information for use over the life cycle.

The concept and connotation of the AST have not yet been unified as the research work is still in its infancy. Systems Engineering Research Center [13] constructs an AST environment by OpenMBEE(Open Model Based Engineering Environment) [14] created by JPL(Jet Propulsion Laboratory). OpenMBEE is an open source collaborative engineering system. It enables engineers to work in the language of their choice and easily share and document their work across other tools. It is aimed at Connected engineering information for a connected world. OpenMBEE has four components. a) Model Development Kits (MDKs) are tool-specific integrations that's primary purposes are to sync models with the Model Management System and implement the DocGen language, which allows modelers to dynamically generate documents in a model-based approach using the view and viewpoint concept. b) Model Management System(MMS) is a version control system for structured data. It exposes model information through RESTful web services that can be used for CRUD(Create, Read, Update, Delete) operations, branching, and tagging of the model repository. c) View Editor(VE) enables users to interact with SysML models within a web-based environment. It implements the MMS REST(Representational State Transfer) API(Application Programming Interface) to provide a web environment to create, read, and update model elements, including Documents and Views. d) OpenSE Cookbook is a compendium for the engineer which captures best practices, lessons learned, and provides guidance on how to use languages and tools to perform a certain engineering task.

Aircraft conceptual design is rapidly iterative and tool-heavy, requiring an AST system that users can use quickly and with less tool coupling. However, OpenMBEE is focused on solving significant MBSE problems, such as the Thirty Meter Telescope. As a complex system, OpenMBEE is difficult to deploy and has a long learning curve, making it inappropriate for the aircraft conceptual design.

Alexander *et al.* [15] summarize four conditions that AST needs to be met. First, AST is recognized by a governing entity as a System of Record (SoR). Second, the majority of experts accept the credibility, accuracy, and trustworthiness of a digital artifact. Third, most experts agree that the source is legitimate. Finally, it maintains its integrity and reinforces the conditions. Sun *et al.* [16] construct an AST system and tested the system through a civil aircraft pre-research design. The construction objectives and principles and the models and data types included in the AST are also proposed. Yue *et al.* [17] construct an effective AST solution by applying blockchain technology.

The gap in this research area is that due to the novelty of the AST concept, there is not yet an AST construction methodology that researchers are figuring out. We need to study the methodology for building AST systems and clarify how to design and develop AST systems that can eventually contribute to the promotion of MBSE in aircraft design.

Our research aims to improve the collaborative design capability, shorten the design cycle, improve the design efficiency, and ensure the accuracy of models and data for the aircraft concept design phase. We designed the AST construction methodology, which focuses on how models and data are collected, managed, and applied. To achieve the research objectives, we study the construction goals of AST, summarize the basic methods of AST construction, and designed the architecture of AST. The study of construction techniques is also part of the methodology, by studying collaborative construction techniques to address model and data acquisition, dynamic management techniques to address management, and basic functions and derivative tools to address application.

In this paper, the AST system is designed based on the Unmanned Combat Aerial Vehicles (UCAV) conceptual design. Compared with manned combat aircraft, UCAV lacks life support systems and has characteristic of low cost. Meanwhile, UCAV is high overload and expandability than manned combat aircraft. Therefore, the conceptual design of UCAV has the characteristics of simplicity and rapidity, which is different from the manned combat aircraft design. In the previous design, UCAV concept design is based on documents, which brings three problems: slow design iterations, inconsistencies, and often rework. MBSE is an intelligent solution to solve the above problems. The AST system can reduce the design cost and with less tool coupling. An AST is a core and base to MBSE, which is closely related to the model and data in the conceptual design of UCAV.

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

1) Design an AST construction methodology, including construction goals and objects, basic methods of construction, the architecture of AST, collaborative construction methods, dynamic management methods, basic functions and derivative tools required.
2) Develop an AST system by applying the AST construction methodology to meet the construction goals and support the UCAV conceptual design.
3) Use the UCAV concept design case to test the usability of the AST system, it is clear that the AST system can help the implementation of MBSE in the field of aircraft design.

This paper is organized as follows. Section II formulates the goal, objectives, and the basic methods of building the AST and designs the architecture of the AST. Then in Section III, the construction techniques of the AST system are studied, including collaborative construction techniques, dynamic management techniques, basic functions, and derivative tools. The composition and usage process of the AST system is also introduced. In Section IV, UCAV's conceptual design case is used to test the feasibility of the AST system, while Section V concludes with benefits and future work.

II. AUTHORITATIVE SOURCES OF TRUTH

The SoR was an early form of the AST. The SoR was used to record any information and elements of the life cycle of the System of Interest (SoI). However, it only makes records but is not certified by most experts for credibility. The AST will provide traceability as the system of interest evolves, capturing historical knowledge, and connecting authoritative versions of models and data. Changes made to the AST will propagate throughout the digital design model to all affected systems and functions. Properly maintaining the AST will mitigate the risk of using inaccurate models and data and effectively control the current and historical configuration data files. The AST can capture the suitable digital models with fidelity and adopt practical management tools to ensure the consistency of digital models. Creators and other stakeholders share these models and data to drive digital transformation jointly.

The AST captures the current state and the history of the technical baseline. It serves as the central reference point for models and data across the life cycle (see Fig. 1). The AST can record models and standardized data for all versions of formalized modeling during the entire life cycle of the SoI. The entire life cycle contains ideas, requirements analysis, architectural design, conceptual design, preliminary design, detailed design, manufacturing, use, maintenance, and end-of-life or recycling.

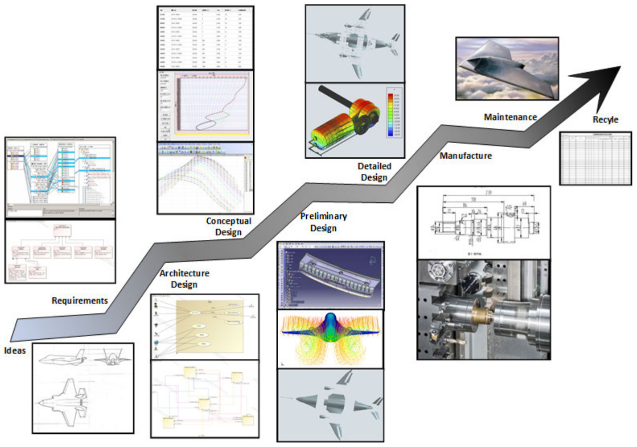


FIGURE 1. Life-cycle models and data.

A model is an abstraction or representation of the system, entity, phenomenon, or process of interest [18]. Models of the system's entire life cycle include technical models and project management models [16], where technical models describe non-geometric models such as mathematical models, physical models, and textual descriptions related to technology and design and geometric models such as 3D models. Project management models are the models related to management in the entire life cycle, such as team models, risk models, schedule models, budget models, and cost models.

The data are accurate and real-time, relied upon and generated throughout the entire life cycle of the SoI. It includes design data, operation data, maintenance data, analysis and reporting data, and reference data.

A. CONSTRUCTION GOAL AND OBJECTS OF THE AST

As the core of MBSE, the AST needs to support the rapid extraction, consistency, tracking and tracing, and V&V (Verification & Validation) of models and data in the design process. Furthermore, the AST should use models and data for continuous and comprehensive digital representation of the system, enhance the efficiency of collaborative work, shorten development time, improve product quality, enhance management efficiency, and support agile R&D (Research And Development). We have a construction goal and five construction objects.

The goal is to record the right models and data, delivering the right models and data to the right stakeholder for the right SE practice at the right time.

Object 1. Improving the models and data reusability. It is one of the goals of MBSE. By replacing the document-based approach with a model-based approach, the AST can help models and data to be able to be used across projects. Object 2. Eliminating inconsistencies. It is also one of the goals of MBSE. After documents are transformed into models, the AST can establish relationships between models and data at different life cycle stages, make comparisons, identify differences, and improve version consistency. Object 3. Providing accurate models and data. Stakeholders' models and data come from the AST, which needs to ensure the credibility of the models and data. Object 4. Reducing complexity. The confusing and interlocking versions of models and data from many sources affect each other, increasing the complexity of collaboration. Creating a unified model and data helps reduce complexity. Object 5. Enhancing management and assisting in decision making. Once models and data are managed in the AST, managers can access data and make decisions quickly and efficiently.

B. BASIC METHODS OF CONSTRUCTION

To ensure that the AST achieves the intended goal and objects, it needs to take specific methods to build. This paper summarizes four methods to build the AST. a) model-based method, b) object-oriented method, c) formalization method, and d) system engineering method. Each method needs to follow its specific principles.

Model-based method. a) Moderate granularity principle. A moderate granularity model is constructed at the conceptual design stage. The model can meet the current usage requirements. b) Interface stability principle. The model should have a correct and stable interface to ensure reusability, and the interface should be straightforward. c) Correctness principle. The model should have the credibility recognized by the stakeholders

Object-oriented method [19]. a) Abstraction. The system can be expressed by a set of more general models with common characteristics. b) Modularity. The model should be decomposed into a set of smaller models that can be managed separately and have complete functionality. c) Encapsulation. The model should be able to be encapsulated as a black box with interfaces for invocation. d) Generalization and inheritance. Models should have the ability to generalize and inherit.

Formalization methods. a) Standards-based. Building the AST should maintain consistency with the industry and institutional standards. For example, ensure that the schema follows a standard modeling language (e.g., SysML, Systems Modeling Language). b) Respect habits. Building the AST should respect engineers' design habits and modeling habits which have their in-memory logic. c) Reduce redundancy. Avoid using redundant information.

Systems engineering method. a) Full life cycle principle. The AST designing needs to consider the entire life cycle, flexible building, and continuously expanding. b) Tailoring principle. The heftier the system size, the more strictly the construct according to the methodology and principles.

C. ARCHITECTURE

1) ARCHITECTURE DESCRIPTION

The architecture of a system represents the essential elements of a system relative to its external environment, e.g., the essence or elements of a system. Architecture describes constituent elements, elements arrangement, organization principles, and evolution principles. In short, architecture refers to the essential organization of a system, encompassing its functions, composition, and relationships. Architecture descriptions are work products of systems architecting. The conceptual model of architecture description (Fig. 2) is used to explain the architecture of the SoI and describe the general concepts in implementing the architecture, the intended use of the system, and the environment. And it is the basis for system design and development activities.

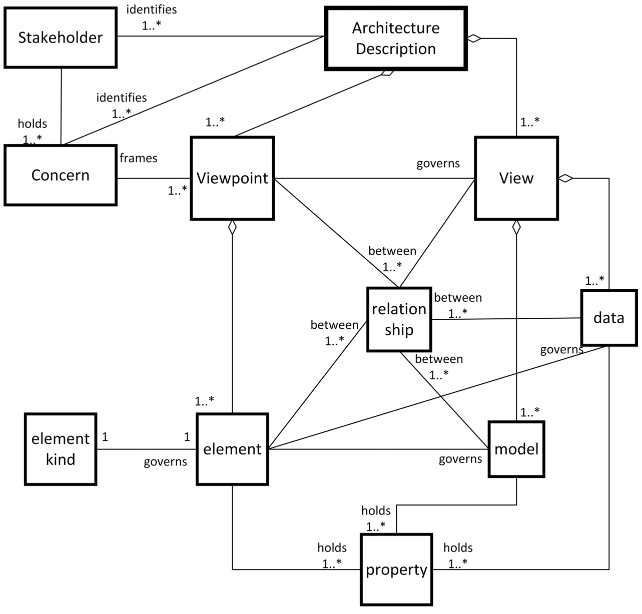


FIGURE 2. The conceptual model of architecture description.

The conceptual composition of the AST architecture description contains stakeholders, stakeholder concerns, viewpoints, views, elements, relationships, models, data, attributes, and element types.

Stakeholder means individual, team, organization, or classes thereof, having an interest in s system.

Concern means interest in a system relevant to one or more of its stakeholders. Concerns relate to any impact of the system in its environment, including developmental, technical, commercial, operational, organizational, political, economic, legal, regulatory, ecological, and social impacts.

Viewpoint means establishing conventions for constructing, interpreting, and using architecture views to frame specific system concerns.

A viewpoint developed from stakeholder concerns represents a design participant's or a design aspect's perception and deconstruction of an AST. A viewpoint is a work product that establishes conventions for constructing, interpreting, and using views. A viewpoint frames one or more concerns. Conventions for views can include language, notation, model types, design rules, modeling methods, analysis techniques, and other manipulations of views.

The view is a work product expressing the architecture of a system from the perspective of specific system concerns. A view expresses the work product of a system architecture from the perspective of a particular system view. Viewpoints govern views(Fig. 3). An architecture description includes one or more views, and the views address one or more issues of concern to stakeholders.

Elements are work products at the smallest granularity, deconstructed from models and data. Each element distincts from others and has unique properties. Element types represent a type unique to each element to distinguish it from other elements. The elements are combined in a certain way to form a model and data, and this combination can be called a relationship.

Relationships are the intrinsic relationships or interrelationships of elements, models, data, views, and viewpoints. Elements and relationships are the most fundamental components of an AST, and elements and relationships combined with specific rules can characterize any model and data in an AST.

Models and data are components of a viewpoint and originate from processes related to the entire life cycle. Models and data are governed by elements, which are collections of elements consisting of certain intrinsic logical relationships. The Attributes are intrinsically unique descriptions of elements, models, and data to distinguish each model and data.

2) ARCHITECTURE OF THE AST

The AST is responsible for the entire life cycle of models and data, covering 30-40 years. Therefore, it has three capabilities: massive, security, and flexibility. Massive means that the AST needs to support massive volume models and data.

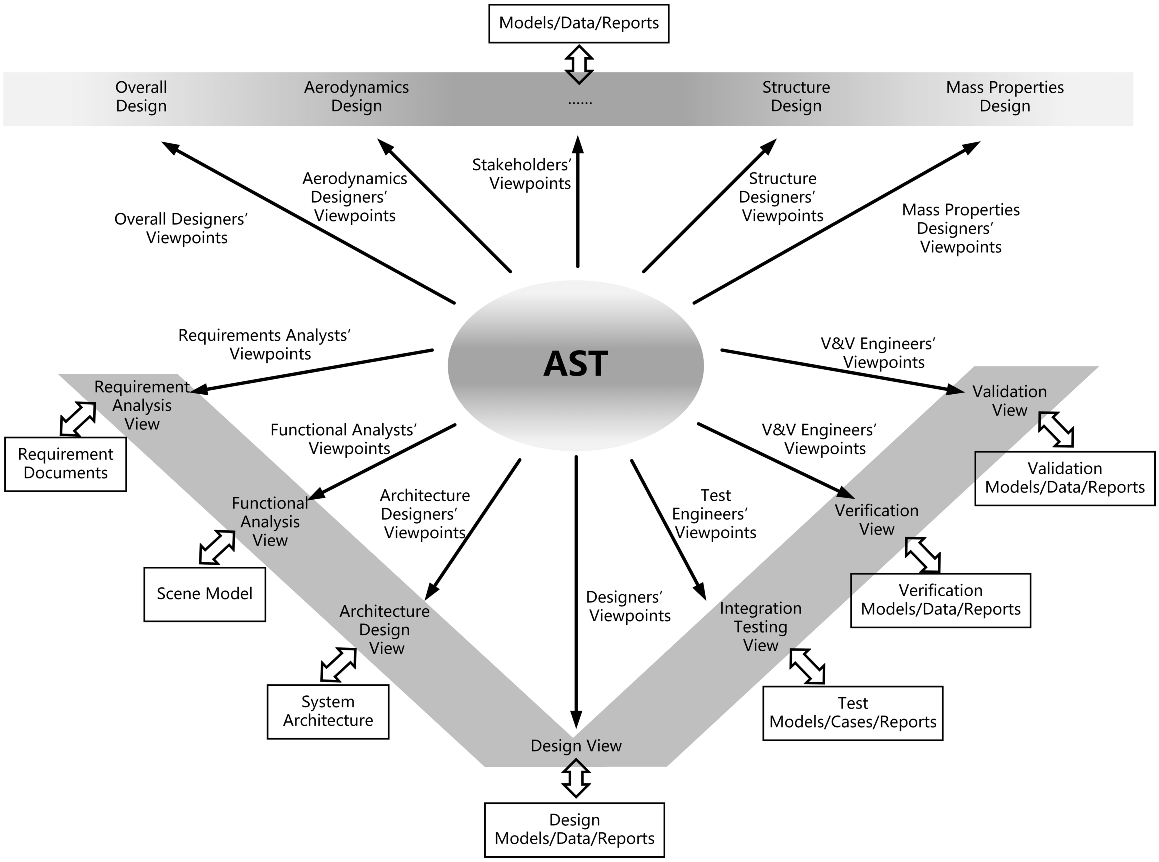

Models/Data/Reports
Overall Design
Aerodynamics Design
......
Structure Design
Mass Properties Design
Overall Designers' Viewpoints
Aerodynamics Designers' Viewpoints
Stakeholders' Viewpoints
Structure Designers' Viewpoints
Mass Properties Designers' Viewpoints
AST
Requirements Analysts' Viewpoints
V&V Engineers' Viewpoints
Requirement Analysis View
Requirement Documents
Functional Analysts' Viewpoints
V&V Engineers' Viewpoints
Validation View
Validation Models/Data/Reports
Functional Analysis View
Scene Model
Architecture Designers' Viewpoints
Test Engineers' Viewpoints
Verification View
Verification Models/Data/Reports
Architecture Design View
System Architecture
Designers' Viewpoints
Integration Testing View
Test Models/Cases/Reports
Design View
Design Models/Data/Reports

FIGURE 3. **The AST, viewpoints, and views.**

Security means that it has distributed storage and backup system. The model and data management have security. Flexibility means that it can be flexibly extended, and the interface is flexible.

The architecture description is used to define the basic concepts and relationships. The architecture is based on the architecture description, an abstract description of entities and their relationships. The architecture is an organic unification of formal structure and functions. The formal structure represents the ontology of the AST and its components, which have been or will be realized. The formal structure is the physical and informational embodiment, which plays an instrumental role in executing the function and is the basis for realizing the function. The functions are the activities, operations, or transformations that the AST can generate or facilitate, the goal, and objects that can support the AST. Designed functions can be executed through the formal structure. Some of the functions have to emerge from the functional interactions between entities.

Fig. 4 shows the architecture of the AST, with the formal structure on the left in the central part, the functions on the right in the central part, configuration management in the upper part, and model & data schema on the lower part. The formal structure includes an infrastructure layer, a service layer, and an application layer. The infrastructure layer refers to the operating system. The service layer contains a middleware support layer, a service construction layer, and the service interface. The models and data of the AST are stored in the service layer and interact with the outside world in the application layer.

The function has an ellipse indicating the procedure and a rectangle indicating the operand. The two-way arrows indicate that the procedure affects the operands, but neither consumes nor creates them. A one-way arrow from the process to the operand indicates that the process creates the operand. The functions include management, upload, access, comparison and analysis, retrieval, cleaning, and visual presentation.

Model & Data Scheme, as a Strong Specification, requires that models and data must be uploaded according to its format. Schema is a cross-tool specification with model and data as the granularity. It specifies what models and data should be submitted by each designer at each stage of the design process, and also specifies the exact format of the models and data. For example, in conceptual sketch phase, Schema includes a conceptual sketch in picture, a description of sketch in text. In the overall layout phase, it includes a layout description in text, a three-sided diagram in figure, a three-dimensional model, a full-set infiltration area in number. It also contains a series of numbers of wing design, fuselage design, and empennage design. Schema is also designed

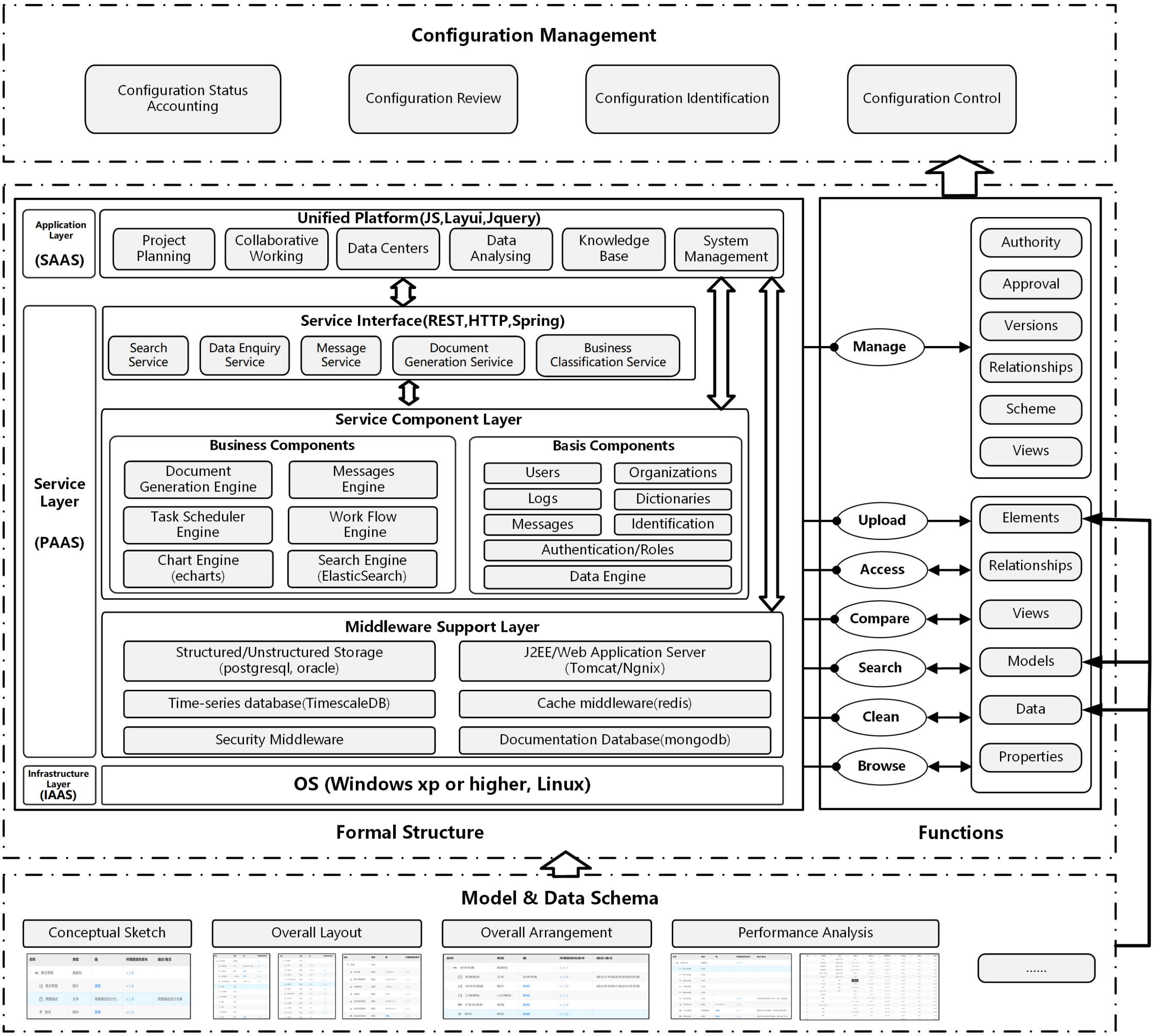


FIGURE 4. Architecture of the AST.

with reserved specifications and attachments for uploading supporting design materials. The benefits of building Schema are not only limited to the normalization of the current project data, but also to the cross-project comparison analysis, which can be done quickly by using the same set of specifications for the model and data.

The AST system supports configuration management for the aircraft conceptual design, a) The process of uploading models and data in the AST system can support configuration status accounting, and the AST system is able to record models and data for aircraft concept design and further record them through the baseline function. b) The approval function supports configuration review and can help check models and data. c) The function of version management supports configuration identification, which can identify each model and data in the AST system. d) The management function supports configuration control, including management authority, models, data, relationships, etc.

The architecture of AST designed in this paper is closely related to the construction goals. The formal structure is used to support functions implementation, while the functions are used to support the goals and objects. Table 1 shows the correspondence between the functions and their goal and objects in the architecture of AST. In this paper, the construction goals are fully satisfied by the rational design of functions. Meanwhile, the formal structure in the architecture is realized by building basic methods, forming a closed-loop design of goals, methods, and architecture.

TABLE 1. Function comparison table.

Functions	Goal and Objects
Authority, Approval	Insure models and data correctness
Versions, Relationships, Schemes, Views	Eliminate inconsistencies and reduce complexity
Upload, Access	Easy access to models and data
Compare, Browse	Improve the efficiency of collaborative decision making
Search	Provide accurate information
Clean	Improves reusability of models and data

III. CONSTRUCTION TECHNIQUES

The AST system is a prototype system designed according to the architecture. The construction techniques are a set of techniques to build an AST system. According to the functions of the architecture design, the construction techniques need to research the collaborative construction techniques, dynamic management techniques, essential functions, and other derived functions. For the multi-dimensional models and data representation of Element → Models/Data → View in the architecture description, the AST system is designed to correspond to it in the form of Element → Data Item → Data Package.

The data item is the same as models and data, which contains models and data with specific intrinsic meaning in the design process. The data package is the smallest unit uploaded to submit to the AST system and consists of data items that stakeholders need to record based on the current phase. The data package is at the same level as the View. They intersect each other. The data package can be either a view or contain multiple views or can currently be contained by a view. A data package is a collection of data items, which means a collection of design data, models, inputs, and outputs related to a particular design effort in the design process.

A. COLLABORATIVE CONSTRUCTION TECHNIQUES

Models and data involve many stakeholders. In collaborative work, they upload and download models and data that need to be done cooperatively in different working environments. Designers and other stakeholders can easily upload, browse, and download data in their work environment, which can reduce unnecessary communication, save time, and improve the efficiency of collaborative work.

The AST system is based on a hybrid B/S + C/S architecture and designed with collaborative construction techniques. Stakeholders are all connected to the AST system through the network, as shown in Fig. 5, which contains two modes of collaborative construction: 1) Accessing the AST through a browser. Firstly, it needs to be verified by permission, and then it can realize the upload and download of models and data in the browser, and it can also extract the models and data in the server for visual display by using the browser. 2) The common tools are different for stakeholders. By secondary development of the common tools and providing their ability to access the AST, stakeholders, especially designers, can upload models and data in the common tools with one click after working, and models and data can be downloaded with one click and directly imported into the common tools for design through pre-defined views.

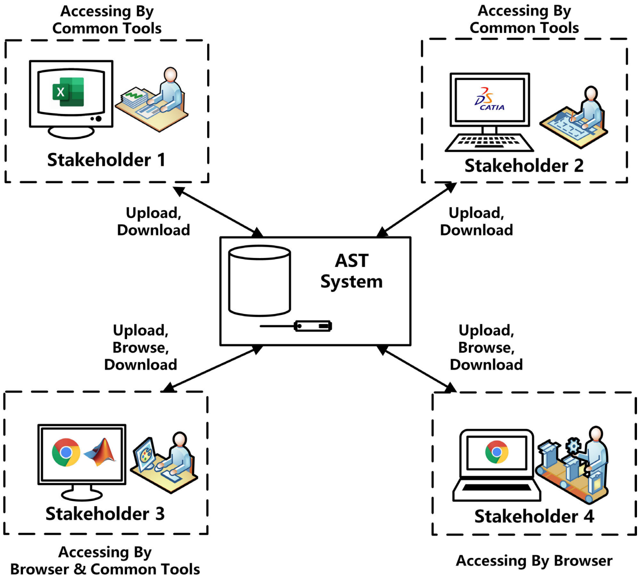


FIGURE 5. Collaborative construction techniques.

B. DYNAMIC MANAGEMENT TECHNIQUES

1) DYNAMIC AUTHORITY MANAGEMENT

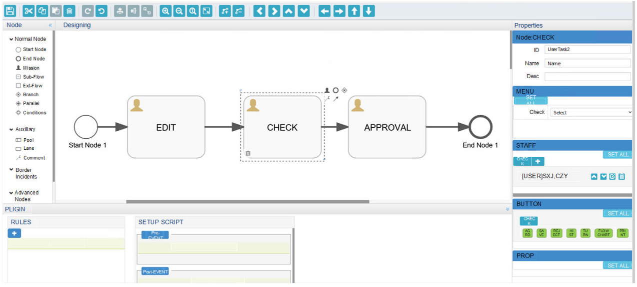
One of the capabilities of the AST is to collect models and data from the right stakeholders and distribute them to the right stakeholders after processing. This paper designs dynamic authority management for the AST system, including permission management and dynamic authorization.

The data package is the smallest unit of a single upload and permission management. Stakeholders can have the responsible permission, download permission, and view permission of the data package. The above three permission ranges are narrowed down in turn. The authority needs to be assigned at the beginning of the project planning according to the role division. The authority of stakeholders in different projects is independent of each other and does not affect each other.

Since the project life cycle process involves a long time, the positions of stakeholders may be transferred and changed, resulting in changes in the authority of the team and members. The dynamic authority technology is designed to keep the project open and not affect the work process of others so that the project team can be managed. The authority of the data package can be modified separately, realizing the dynamic adjustment of authority without any sense of the user.

2) DYNAMIC APPROVAL MANAGEMENT

The AST needs to guarantee the credibility of models and data. We design the approval management, where the models and data in the data package that pass the approval process completely can guarantee credibility. The designed approval



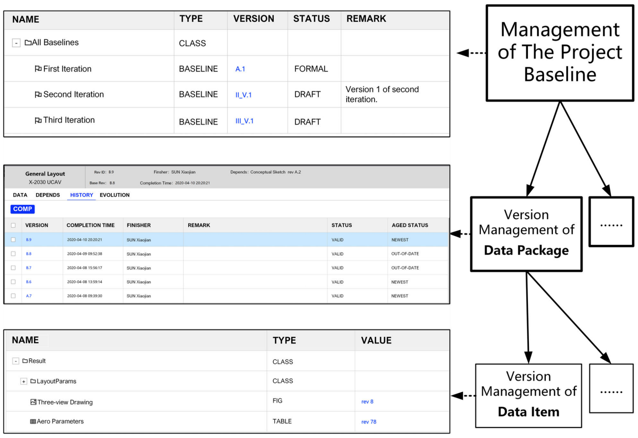
ID UserTask2
Name Name
Desc
Check Select
[USER]SXJ,CZY

FIGURE 6. Approval process.

process (Fig. 6) includes the number of nodes for approval, personnel of each approval node, approval results, and processing mechanism. Different data packages have different working scenarios. By pre-configuring the approval process, the number of nodes and personnel rights are flexibly set to meet the approval in all states to ensure the model's credibility and data. This paper designs the cancellation function which allows the submitter cancels the process at any stage of the approval.

3) DYNAMIC VERSION MANAGEMENT

Version management refers to the process of recording, tracking, and controlling changes to models and data. The dynamic version management contains project baseline management, data package version management, and data item version management, as shown in Fig. 7. The dynamic version management can help the model and data versions of different professions, and stages in the design process correspond and are consistent.



NAME	TYPE	VERSION	STATUS	REMARK
- All Baselines	CLASS			
First Iteration	BASELINE	A.1	FORMAL	
Second Iteration	BASELINE	II_V.1	DRAFT	Version 1 of second iteration.
Third Iteration	BASELINE	III_V.1	DRAFT	

General Layout
X-2030 UCAV
Rev ID：B.9
Base Rev：B.8
Finsher：SUN Xiaojian
Completion Time：2020-04-10 20:20:21
Depends：Conceptual Sketch rev A.2

□	VERSION	COMPLETION TIME	FINISHER	REMARK	STATUS	AGED STATUS
□	B.9	2020-04-10 20:20:21	SUN Xiaojian		VALID	NEWEST
□	B.8	2020-04-09 09:52:38	SUN Xiaojian		VALID	OUT-OF-DATE
□	B.7	2020-04-08 15:56:17	SUN Xiaojian		VALID	OUT-OF-DATE
□	B.6	2020-04-08 13:59:14	SUN Xiaojian		VALID	NEWEST
□	A.7	2020-04-08 09:39:30	SUN Xiaojian		VALID	NEWEST

NAME	TYPE	VALUE
- Result	CLASS	
+ LayoutParams	CLASS	
Three-view Drawing	FIG	rev 8
Aero Parameters	TABLE	rev 78

Management of The Project Baseline
Version Management of Data Package
......
Version Management of Data Item
......

FIGURE 7. Version management of data item, data package, and baseline.

Projects can be created in the AST, distinguished, and managed by different project names. Baselines represent the evolution of the project at different times. Furthermore, baselines can select some of the data packages and data items for release. Each of them retains the versions of the data packages and data items at the time of release. Baselines can be modified before release to change the versions of data packages and data items.

Data packages can be dynamically changed according to the entire life cycle development. Each data package is centrally managed and generates a unique version number, and its version is dynamically updated every time the data package is uploaded. The version information contains a version number, completion person, completion time, package notes, and attachments.

Similar to the data package, a unique version number exists for each data item. Every time a data package is updated, the system automatically compares the differences between data items and generates a unique version number for the data items that have changed.

Version management can help stakeholders to manage traceability. It means tracing the source and destination of any model or data, including the upstream tracing of the data package, the baseline tracing of the development of the data package itself, the evolutionary history of the model and data, and the changes of the model and data among different versions.

4) DYNAMIC RELATIONSHIP MANAGEMENT

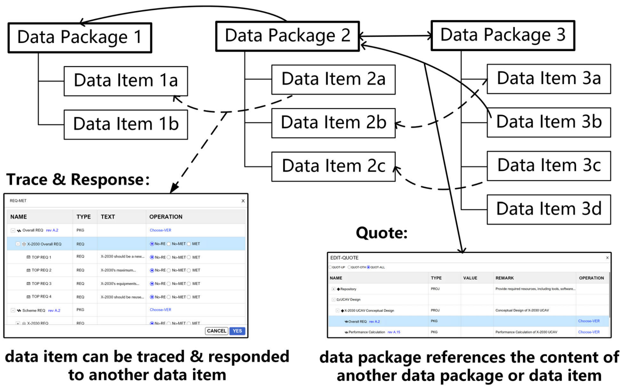
According to the conceptual model of architecture description, there are numerous relationships. In addition to the relationship between the models and data, the AST system also has a data package relationship and a data item relationship, and there is a coupling effect between the two relationships. In this paper, we design two kinds of relationships, Quote and Trace & Response.

The quote relationship represents that a data package references the content of another data package or data item. It is a weak relationship established by the submitter of the data package and used to draw the attention of stakeholders and maintain version consistency with dynamic version management.

Trace & Response is a stronger coupling relationship, representing that a data item can be traced and responded to another data item. It is established to track and record the changes of both data items, such as the traceback response between requirements, as shown in the Fig. 8. The dynamic nature of relationship management is reflected in the fact that the two relationships can be constantly modified and improved according to the actual situation during the development of the time course. The description of the relationship is constantly accurate.

5) DYNAMIC SCHEME MANAGEMENT

The dynamic scheme management technique is also designed to ensure project consistency, with partial model support for reuse between multiple schemes, but no mixing of models and data should exist. The dynamic scheme management technique designed in this paper uses a combination of evolution and derivation. For a particular data package under the responsibility of a stakeholder, dynamic evolution occurs continuously within the same scheme, iterating forward while



REQ-MET

NAME	TYPE	TEXT	OPERATION
Overall REQ rev A.2	PKG		Choose-VER
X-2030 Overall REQ	REQ		● No-RE ○ No-MET ○ MET
TOP REQ 1	REQ	X-2030 should be a new...	● No-RE ○ No-MET ○ MET
TOP REQ 2	REQ	X-2030's maximum...	● No-RE ○ No-MET ○ MET
TOP REQ 3	REQ	X-2030's equipments...	● No-RE ○ No-MET ○ MET
TOP REQ 4	REQ	X-2030 should be reuse...	● No-RE ○ No-MET ○ MET
Scheme REQ rev A.2	PKG		Choose-VER
X-2030 REQ	REQ		● No-RE ○ No-MET ○ MET

CANCEL YES

EDIT-QUOTE

○ QUOT-UP ○ QUOT-OTH ● QUOT-ALL

NAME	TYPE	VALUE	REMARK	OPERATION
Repository	PROJ		Provide required resources, including tools, software...	
UCAV Design				
X-2030 UCAV Conceptual Design	PROJ		Conceptual Design of X-2030 UCAV	
Overall REQ rev A.2	PKG			Choose-VER
Performance Calculation rev A.15	PKG		Performance Calculation of X-2030 UCAV	Choose-VER

FIGURE 8. Quote for data package and trace & response for data item.

allowing the package to be dynamically derived at the right time, branching out from the baseline scheme to cope with multiple scenarios.

As shown in Fig. 9, the process of evolution and derivation of scheme V is shown, where versions V.8 and V.9 are the results of evolution, and scheme A (including versions A.1, A.2, A.3), scheme B (including version B.1), scheme C (including versions C.1, C.2), and scheme D (including version D.1) are the result of derivation. The status of each version is either VALID or NEW-VALID. NEW-VALID means this is the latest iteration version with a green color indication. VALID indicates that this is an evolved version and represents only the validity of the result, not the timeliness.

6) DYNAMIC VIEW MANAGEMENT

Because the data package can only represent a specific single view of the stakeholders, many views cannot be displayed explicitly, limiting the stakeholder approach's extraction of models and data. This paper designs dynamic view management techniques to help implement custom views and dynamic view sharing.

The views can remove redundant information and be customed according to the models and data displayed requirements. They are templates of models and data. Contents of views depending on the concerns and views of stakeholders, and different views can build data packages (one or more) into different views. For the same package, different designers care about different data. Depending on respective viewpoints, a view can be constructed by selecting from the package in advance. The AST system supports users to export views containing models and data to facilitate daily work, in case that the models and data can be quickly extracted.

View sharing is also designed to share their self-built views with other stakeholders who have the exact needs. It helps reduce the amount of collaborative work.

C. BASIC FUNCTION

1) FLEXIBLE EXTENSION

During the development of the entire life cycle of interest systems, the number and the variety of models and data are expanding. In this paper, flexible extension techniques are designed to help enhance management and assist decision-making. The flexible extension technique includes the flexible extension of the element type, data item, and data package.

The element type is the basis of the model and data, and too many kinds of models and data are involved in the project's entire life cycle. Even if it is detailed to the element, its type is dynamic change and continuous expansion, which can be flexibly expanded according to the different stages of the project. The current element types of the AST system are text, values, curves, tables, CAD models, CAE models, figures, audio and video, web pages, appendix, and requirements. Corresponding to the element type, the data item type are TEXT, NUM, CURVE, TABLE, CAD, CAE, FIG, AV, WEB, APPENDIX, and REQ. Currently, SysML models are uploaded to the AST system in web pages and attachments by exporting SysML models to web page format in MagicDraw or CSM (Cameo Systems Modeler).In the future, it will be possible to expand types such as multi-dimensional parameter tables and the SysML Models, and the expanded types will have no impact on the existing types.

In different life cycle stages, the same data package of different versions have changes in contents, quantities, or orders of data items. A flexible extension technique for data items is designed so that data items in a data package can be deleted or removed at one submission. At the same time, the flexible extension will not affect the models and data in other data items. The versioning of data items is based on their IDs, bound to the name and type of the data item. The invariance of ID allows comparison analysis through versioning.

In different life cycle stages, the data packages will be added or removed. In this paper, we design a flexible expansion technique for data packages, which allows the data packages to be added, deleted, and switched back and forth in the project without affecting the data packages that have already been submitted. This technique ensures that the requirements of the project development stage can be met without affecting the trustworthiness of the model and data. As shown in the three parts of Fig. 10, the left side shows the element types, data items, and data packages before the flexible expansion, and the right side shows the element types, data items, and data packages after the flexible expansion.

2) DATA CLEAN

Data clean is an essential means to improve model reusability, including backend cleaning and interface opening. After the models and data are submitted in data packages, they are stored in data items. Data cleaning occurs in the backend. In this paper, cleaning rules are designed for each element; curve type can clean out the table header and each column of data, table type can clean out each cell of data, value type can clean out the data and units, CAD model type can clean out the structure tree, CAE type can clean out the layers and surface elements, requirement type can clean out the requirement

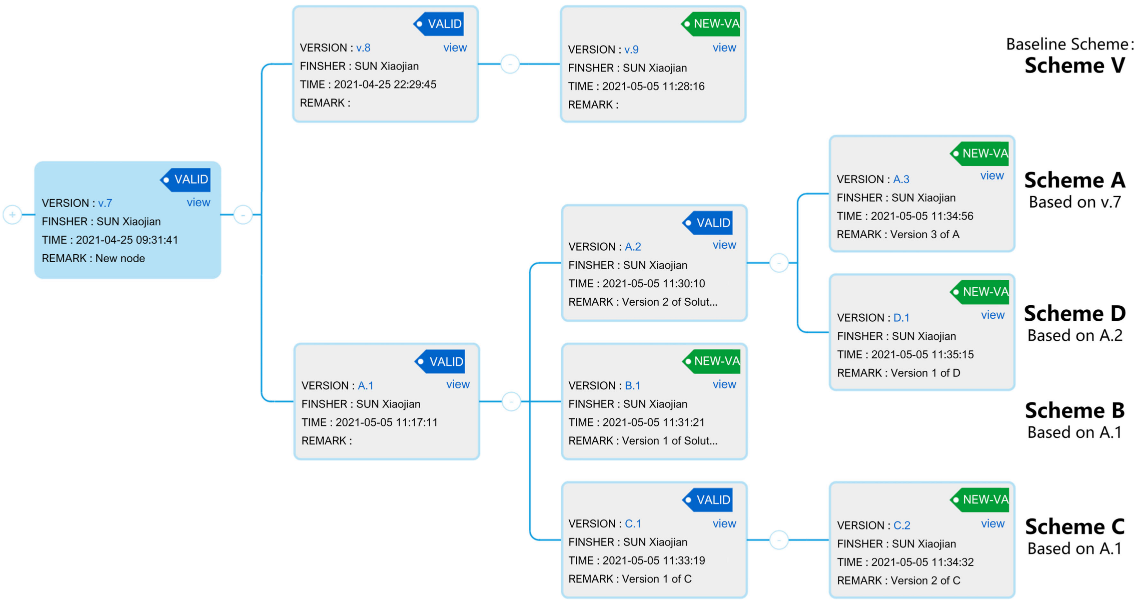
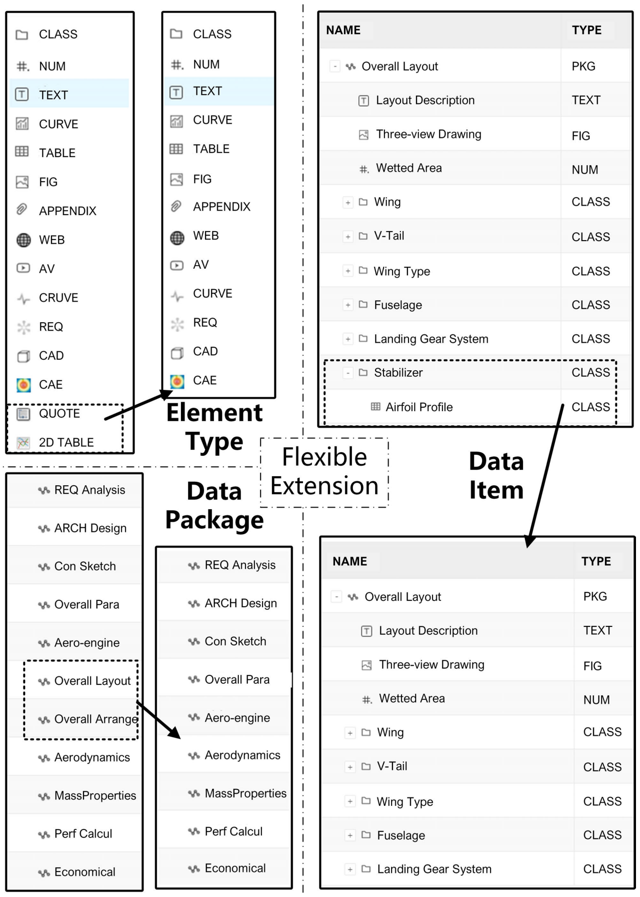


FIGURE 9. The process of evolution and derivation of scheme V.



NAME	TYPE
Overall Layout	PKG
Layout Description	TEXT
Three-view Drawing	FIG
Wetted Area	NUM
Wing	CLASS
V-Tail	CLASS
Wing Type	CLASS
Fuselage	CLASS
Landing Gear System	CLASS
Stabilizer	CLASS
Airfoil Profile	CLASS

NAME	TYPE
Overall Layout	PKG
Layout Description	TEXT
Three-view Drawing	FIG
Wetted Area	NUM
Wing	CLASS
V-Tail	CLASS
Wing Type	CLASS
Fuselage	CLASS
Landing Gear System	CLASS

FIGURE 10. Flexible extension of data item, data package, and baseline.

ID, requirement text, and requirement attributes. The future extension of the multi-dimensional parameter table can clean out the combined parameters of all dimensions.

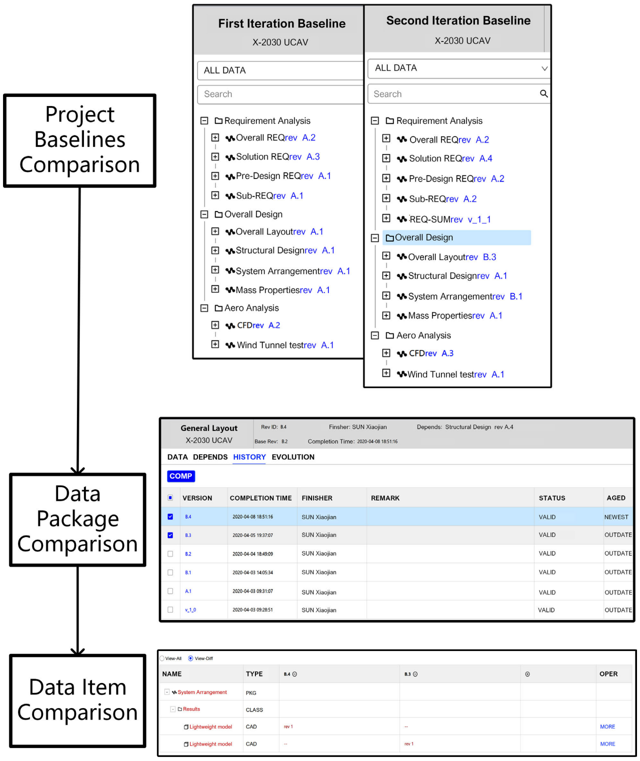
TABLE 2. Customized JSON format of AST system.

JSON Fields	Definitions
Version	The Version of Data Item
ParmID	ID of Data Item
ParmName	Name of Data Item
ParmType	Type of Data Item (Internal convention code)
Value	Value of Data Item (text, values, table contents, curve contents, relative paths for images.)
Remark	Remark of Data Item
Unit	Unit of Values
Sort	Order of Data Items in the Data Package
VersionFlag	Version Description
Children	Content of Sub-data Items

The AST system provides an open external interface. Stakeholders can compile programs based on the interface and directly download the models and data. The models and data are provided in JSON format(Table 2), which is easy to read and write by humans and easy to parse and generate by machines, and effectively improves network transmission efficiency, enhances model and data reusability, and reduces complexity.

3) COMPARISON ANALYSIS

The comparison analysis can compare the differences between different versions. The techniques improve version consistency and collaborative decision-making efficiency, reduce complexity, and include comparison of project baseline, data package, and data item. The comparison analysis compares data on an element-by-element basis to obtain the differences between versions. Fig. 11 shows the comparison between baseline, data package, and data item versions. The project baseline comparison can differentiate baselines versions, containing the number and content of data packages



First Iteration Baseline
X-2030 UCAV
ALL DATA
Search
- Requirement Analysis
  - Overall REQrev A.2
  - Solution REQrev A.3
  - Pre-Design REQrev A.1
  - Sub-REQrev A.1
- Overall Design
  - Overall Layoutrev A.1
  - Structural Designrev A.1
  - System Arrangementrev A.1
  - Mass Propertiesrev A.1
- Aero Analysis
  - CFDrev A.2
  - Wind Tunnel testrev A.1

Second Iteration Baseline
X-2030 UCAV
ALL DATA
Search
- Requirement Analysis
  - Overall REQrev A.2
  - Solution REQrev A.4
  - Pre-Design REQrev A.2
  - Sub-REQrev A.2
  - REQ-SUMrev v_1_1
- Overall Design
  - Overall Layoutrev B.3
  - Structural Designrev A.1
  - System Arrangementrev B.1
  - Mass Propertiesrev A.1
- Aero Analysis
  - CFDrev A.3
  - Wind Tunnel testrev A.1

General Layout
X-2030 UCAV
Rev ID: B.4 Finsher: SUN Xiaojian Depends: Structural Design rev A.4
Base Rev: B.2 Completion Time: 2020-04-08 18:51:16
DATA DEPENDS HISTORY EVOLUTION
COMP

■	VERSION	COMPLETION TIME	FINISHER	REMARK	STATUS	AGED
☑	B.4	2020-04-08 18:51:16	SUN Xiaojian		VALID	NEWEST
☑	B.3	2020-04-05 19:37:07	SUN Xiaojian		VALID	OUTDATE
□	B.2	2020-04-04 18:49:09	SUN Xiaojian		VALID	OUTDATE
□	B.1	2020-04-03 14:05:34	SUN Xiaojian		VALID	OUTDATE
□	A.1	2020-04-03 09:31:07	SUN Xiaojian		VALID	OUTDATE
□	v_1_0	2020-04-03 09:28:51	SUN Xiaojian		VALID	OUTDATE

○ View-All ● View-Diff

NAME	TYPE	B.4 ⊖	B.3 ⊖	⊕	OPER
System Arrangement	PKG				
Results	CLASS				
Lightweight model	CAD	rev 1	--		MORE
Lightweight model	CAD	--	rev 1		MORE

FIGURE 11. Comparison of data item, data package, and baseline.

and the number and content of data items. The data package comparison can get the number and content difference of data items between different versions. The data item comparison can get the difference of content between different versions.

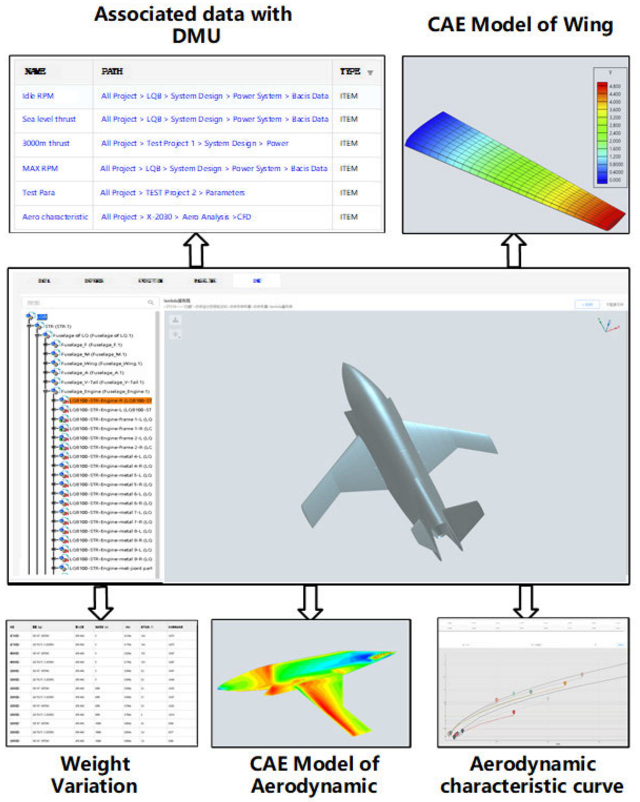
4) DATA SEARCH

Quickly retrieving and extracting key models and data from the massive models and data stored in the AST system is one of the essential techniques.

The data search techniques designed in this paper are based on indexing. Data are indexed by projects, data packages, and data items to build intelligent retrieval methods. Users perform an intelligent search by 1) search by baseline, data packages, and data items, 2) search by type of elements, 3) search and filtering by completion person and completion time, 4) search by tags formed manually during the submission, and 5) search based on the knowledge graphs.

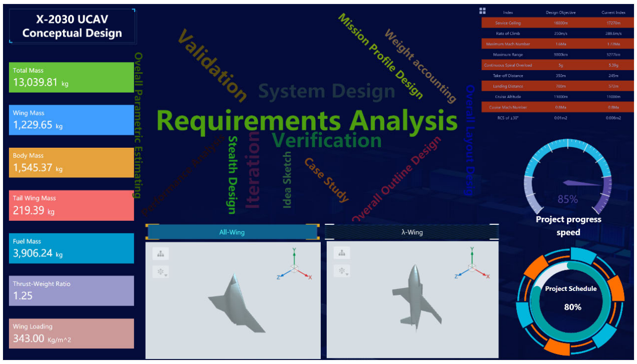
5) VISUALIZATION

The visualization techniques are based on the view, which extracts the model and data from the data package and displays its elements. Displaying directly in the AST system can reduce the complexity of opening locally after downloading. It also can keep the model consistency, reduce the frequency of wrong displays caused by different personal working environments, and improve the efficiency of collaborative work.



NAME	PATH	TYPE
Idle RPM	All Project > LQB > System Design > Power System > Bacis Data	ITEM
Sea level thrust	All Project > LQB > System Design > Power System > Bacis Data	ITEM
3000m thrust	All Project > Test Project 1 > System Design > Power	ITEM
MAX RPM	All Project > LQB > System Design > Power System > Bacis Data	ITEM
Test Para	All Project > TEST Project 2 > Parameters	ITEM
Aero characteristic	All Project > X-2030 > Aero Analysis >CFD	ITEM

FIGURE 12. Visualization for DMU.



Index	Design Objective	Current Index
Service Ceiling	16000m	17270m
Rate of Climb	250m/s	289.6m/s
Maximum Mach Number	1.6Ma	1.72Ma
Maximum Range	3000km	3277km
Continuous Spiral Overload	5g	5.39g
Take-off Distance	350m	245m
Landing Distance	700m	572m
Cruise Altitude	11000m	11000m
Cruise Mach Number	0.8Ma	0.8Ma
RCS of ±30°	0.01m2	0.006m2

FIGURE 13. Monitoring panel.

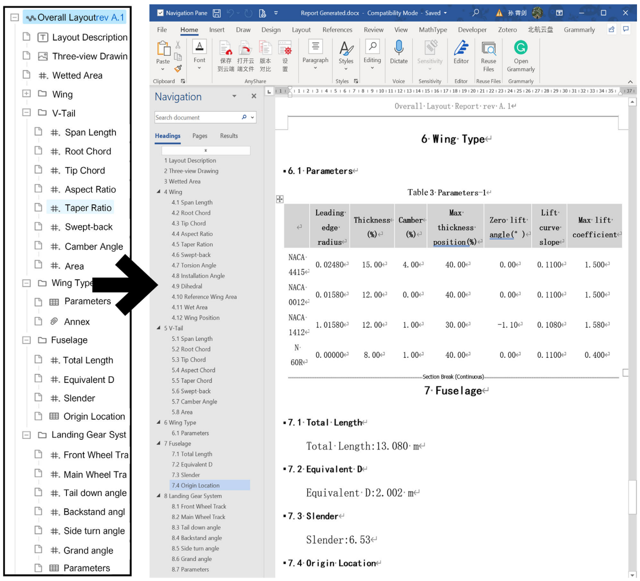
According to the needs of the data items types, the AST system supports multiple visualizations of pictures, audio and video, curves, CAD models, and CAE models.

This paper takes a Digital Mock-Up (DMU) to demonstrate the visualization techniques for the entire life cycle. The DMU is organically linked with the product's performance data, as shown in Fig. 12.

D. DERIVATIVE TOOLS

1) MONITORING PANEL

With the real-time models and data submitted by different stakeholders, this paper designs a monitoring panel to display



Overall Layout Report rev A.1

6 Wing Type

6.1 Parameters

Table 3 Parameters-1

	Leading edge radius	Thickness (%)	Camber (%)	Max thickness position(%)	Zero lift angle(°)	Lift curve slope	Max lift coefficient
NACA 4415	0.02480	15.00	4.00	40.00	0.00	0.1100	1.500
NACA 0012	0.01580	12.00	0.00	40.00	0.00	0.1100	1.500
NACA 1412	1.01580	12.00	1.00	30.00	-1.10	0.1080	1.580
N 60R	0.00000	8.00	1.00	40.00	0.00	0.1100	0.400

7 Fuselage

7.1 Total Length

Total Length:13.080 m

7.2 Equivalent D

Equivalent D:2.002 m

7.3 Slender

Slender:6.53

7.4 Origin Location

FIGURE 14. Automatic report generation.

the project's current progress in real-time as Fig. 13. The monitoring panel is also designed flexibly, and the content and number of panels can be added or deleted according to the needs of different scenarios. The panel can display all element types of the AST system and use various charts to assist in decision-making and improve collaboration efficiency.

2) AUTOMATIC REPORT GENERATION

A key feature of MBSE is the ability to use models for automatic report generation, saving designers time and improving collaboration efficiency.

Reports can quickly generate with custom views. The view provides the scope of report generation, and the scope of the report can be personalized by customizing the view to encompass different models and data. The hierarchy of data items in the view is responsible for building the skeleton of the report, i.e., the headings of each level of the customized report. The models and data are the flesh and blood of the report. The models and data are filled into different positions according to the needs of the report template, and beautiful images are automatically generated and inserted into the report through visual display technology to form a customized design report, as shown in Fig. 14. The automatically generated report can adjust the format, design the template as needed, change the cover, and add headers and footers.

E. COMPOSITION AND USAGE PROCESS

The AST system is the ground implementation of the construction techniques and the specific scenario for conducting UCAV design. According to the form structure of architecture, the system consists of a Home Page, Project Planning, Collaborative Work, Data Center, Personal Center, System Management, Knowledge Base, as shown in Fig. 15. The function of the Home Page is a comprehensive display of information, which contains four parts: the display of pending items, items to be approved, current projects, and message notification. The Personal Center can display and operate the pending and approved items involving individuals in the project centrally. The System Management provides the basis for the whole system, and each manages human resources, project progress and also maintains the primary data involved in project work, such as setting the unit system.

The Project Planning uses dynamic management techniques for planning, building, assigning human resources, and permissions to projects. The Collaborative Working function submits models and data for review by project participants and is essential for building the AST. The Data Center is the most core component, storing all models and data for centralized display of projects with viewing privileges, and then for further display and operation, which contains data that is authoritative data and models.

The Knowledge Base stores detailed information on many aircrafts and aircraft-related equipments and can be continuously expanded to reference project design. When using the AST system, users log in to their accounts through the Home Page, plan projects in the Project Planning, collaborate on projects in the Collaborative Work, then submit models and data into the Data Center, and work through the Personal Center, System Management, and Knowledge Base.

IV. UCAV CONCEPTUAL DESIGN CASE STUDY

A. OVERVIEW

X-2030 UCAV takes the future war in 2030 as the main scenario, highlights the capability characteristics of UAV and fighter aircraft in cooperative air warfare, and carries out the conceptual design of cooperative UAV. The design of X-2030 UCAV requires the design analysis of general arrangement, aerodynamic layout, stealth characteristics, weight balance, flight performance, operation and stability characteristics, structural form, critical systems. It requires feasibility analysis, index iteration, and finally forms a cooperative UAV solution with self-consistent requirements and solutions.

The X-2030 UCAV project runs through the entire process of the aircraft conceptual design, which includes more disciplines. At the same time the project has complete models and data, with sufficient verification basis, suitable as a case to verify the feasibility of the AST system.

B. X-2030 CO-DESIGN MODEL

The co-design model of X-2030 UCAV is based on the systems engineering V model, as shown in Fig. 16, which includes requirement analysis, functional architecture design, scheme design, iteration, and V&V. The scheme design part involves more models and data than the other part. It is also based on the V model, which can be refined into mission profile design, conceptual sketch, overall parameters estimation, overall layout design, overall arrangement design, stealth design, performance analysis, multi-scheme evaluation, and mass accounting. The collaborative construction of X-2030

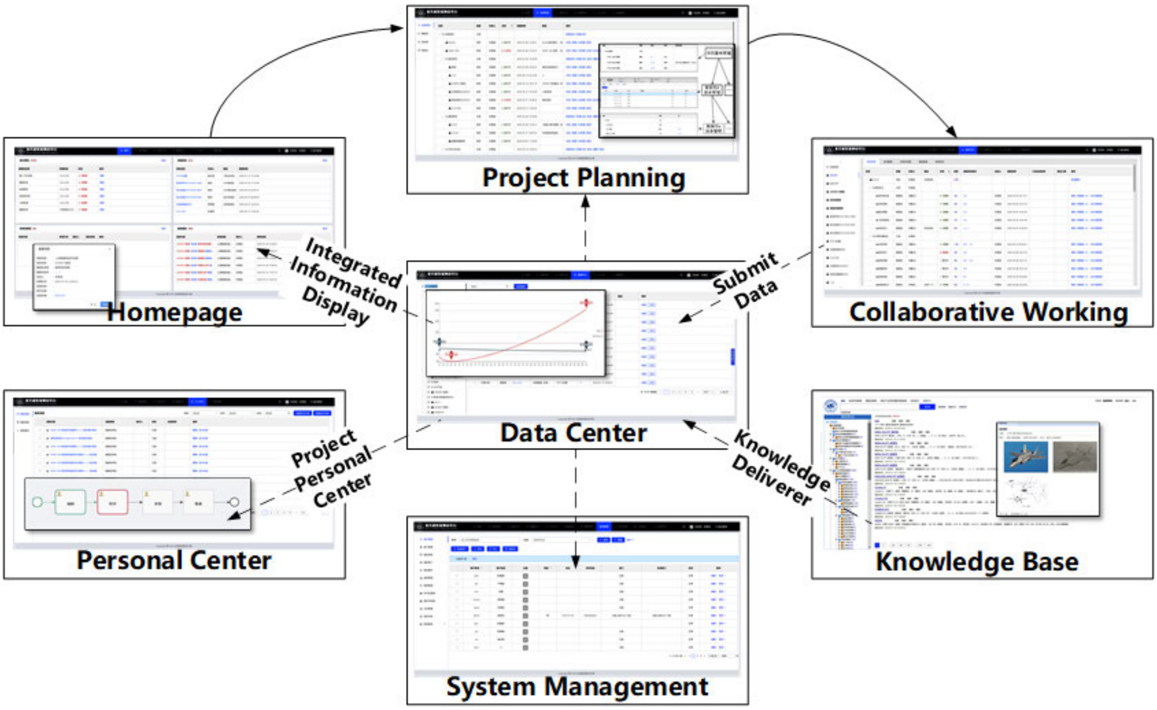


FIGURE 15. Composition and usage process of the AST system.

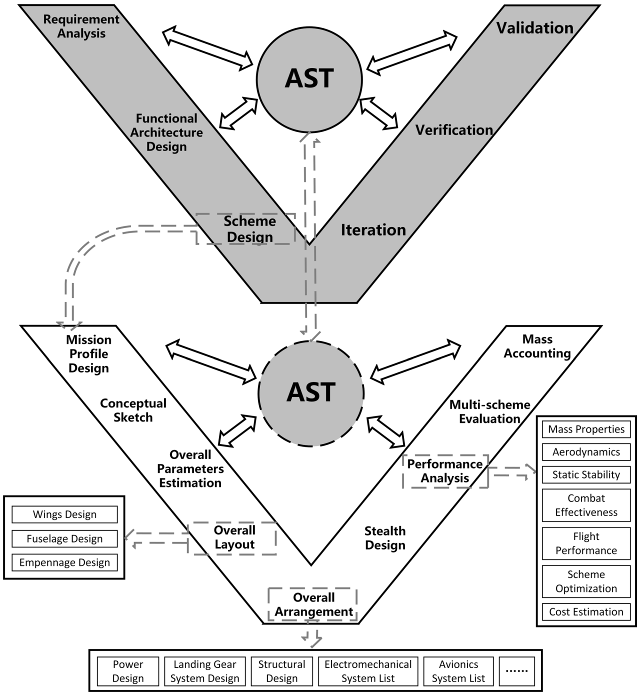


FIGURE 16. Co-design model of X-2030.

UCAV takes the AST system as the digital thread through the stakeholders' upload, view, and download the model data, which ensures the credibility of the model and data and the consistency of the version during the iteration process.

In this case, the stakeholders include requirement analysts, architecture designers, and designers. Requirement analysts are responsible for requirement analysis and requirement validation; architecture designers are responsible for functional architecture design and solution verification; designers are responsible for solution design and solution iteration.

1) REQUIREMENT ANALYSIS

The requirement Analysis was based on the design scenario and capability characteristics of X-2030 UCAV. The relationship between requirements was established, and four layers of requirements were constructed. The four layers, overall requirements, solution requirements, preliminary design parameters, and subsystem requirements, are totaled 68 system requirements and more than 200 subsystem requirements. Fig. 17 shows the overall requirements. Following the guideline that the model only needs to meet the current design stage, the requirements are no longer decomposed in detail and have already met the design scenarios and capability characteristics.

2) FUNCTIONAL ARCHITECTURE DESIGN

The functional architecture design of X-2030 UCAV is based on the results of requirements analysis. The functional architecture design uses SysML to build the architecture model to make each component and subsystem meet the requirements in the logical domain. Functional architecture design can form a closed loop by defining interfaces and simulating the logic inside the system. Problems and unexpected situations

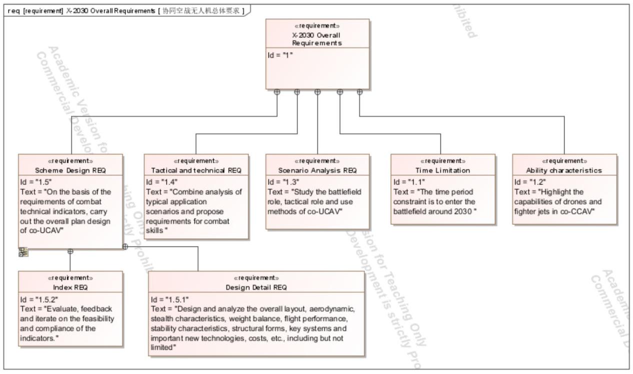

req [requirement] X-2030 Overall Requirements [协同空战无人机总体要求]
«requirement»
X-2030 Overall Requirements
Id = "1"
«requirement»
Scheme Design REQ
Id = "1.5"
Text = "On the basis of the requirements of combat technical indicators, carry out the overall plan design of co-UCAV"
«requirement»
Tactical and technical REQ
Id = "1.4"
Text = "Combine analysis of typical application scenarios and propose requirements for combat skills "
«requirement»
Scenario Analysis REQ
Id = "1.3"
Text = "Study the battlefield role, tactical role and use methods of co-UCAV"
«requirement»
Time Limitation
Id = "1.1"
Text = "The time period constraint is to enter the battlefield around 2030 "
«requirement»
Ability characteristics
Id = "1.2"
Text = "Highlight the capabilities of drones and fighter jets in co-CCAV"
«requirement»
Index REQ
Id = "1.5.2"
Text = "Evaluate, feedback and iterate on the feasibility and compliance of the indicators."
«requirement»
Design Detail REQ
Id = "1.5.1"
Text = "Design and analyze the overall layout, aerodynamic, stealth characteristics, weight balance, flight performance, stability characteristics, structural forms, key systems and important new technologies, costs, etc., including but not limited"

FIGURE 17. Overall requirements of X-2030.

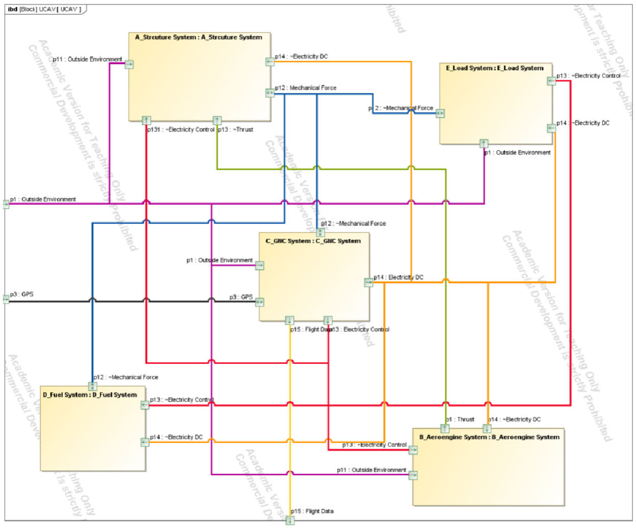

ibd [Block] UCAV [UCAV]
A_Strcuture System : A_Strcuture System
p11 : Outside Environment
p14 : ~Electricity DC
p12 : Mechanical Force
p131 : ~Electricity Control
p13 : ~Thrust
E_Load System : E_Load System
p13 : ~Electricity Control
p 2 : ~Mechanical Force
p14 : ~Electricity DC
p1 : Outside Environment
p1 : Outside Environment
p12 : ~Mechanical Force
C_GNC System : C_GNC System
p1 : Outside Environment
p3 : GPS
p14 : Electricity DC
p3 : GPS
p15 : Flight Data
p13 : Electricity Control
p12 : ~Mechanical Force
D_Fuel System : D_Fuel System
p13 : ~Electricity Control
p14 : ~Electricity DC
p1 : Thrust
p14 : ~Electricity DC
B_Aeroengine System : B_Aeroengine System
p13 : ~Electricity Control
p11 : Outside Environment
p15 : Flight Data

FIGURE 18. X-2030 UCAV IBD.

that arise during the design process can be solved in the logical domain, avoiding problems in the physical domain that cause unnecessary time loss and economic loss.

This paper designed the system-level IBD (Internal Block Definition diagram) of X-2030 UCAV (Fig. 18), which defines the interface relationships among structure system, aero-engine system, GNC system, fuel system, and load system, and defines many activity diagrams (Fig. 19) according to the mission scenarios. The activity diagram is derived from the use case diagram, which includes preflight check, takeoff preparation, recovery check, and mission execution. The mission execution activities include collaborative escort, collaborative interception, and collaborative combat. The IBD and activity diagram of the SysML model is uploaded to AST System in web pages and attachments. AST system supports web pages and *.mdzip source files and can upload, download, manage, browse, and access. The current problem is that the AST system cannot parse SysML models. We can only manage them temporarily in this way. The function of parsing SysML language and *.mdzip is under development and will be released in the next version of the AST System.

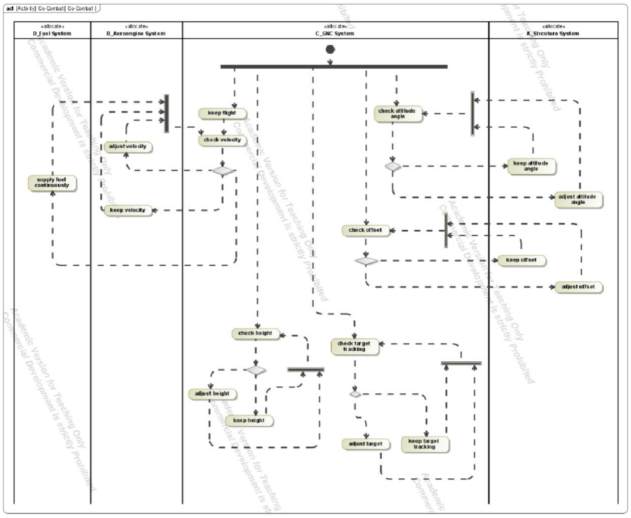


FIGURE 19. X-2030 UCAV collaboration combat activity diagram.

TABLE 3. Closing the loop on design index.

Index	Design Index	Design Result
Practical lift limit	16000m	17270m
Climb rate	250m/s	289.6m/s
Maximum Mach number	1.6Ma	1.72Ma
Maximum Range	3000km	3277km
Continuous circling overload	5g	5.39g
Takeoff and landing distance	350/700m	245/572m
Cruising altitude	11000m	11000m
Cruise Mach number	0.8Ma	0.8Ma
Headway RCS	0.01m^2	0.006m^2

3) SCHEME DESIGN

Based on the X-2030 requirement analysis and functional architecture design, the scheme design is launched. The overall layout design includes wing design, fuselage design, and tail design; the overall arrangement design includes power system design, landing gear design, structural design, avionics system list, and electromechanical system list; the performance analysis includes weight characteristics, aerodynamic characteristics, flight performance, program selection, static stability analysis, operational effectiveness assessment, and cost estimation. The AST system manages the design results, which are shown in Fig. 20. This figure covers the entire process of program design, and the screenshots of the design results shown in the figure come from the AST system. The models and data in Fig. 20 generated at each step are submitted and downloaded through the AST system to achieve traceability and consistency of models and data.

After iteration, verification, and validation, the overall design of X-2030 UCAV meets the requirements analysis results, and the design metrics are closed in Table 3.

C. X-2030 DYNAMIC MANAGEMENT

The dynamic management of the AST system supports the X-2030 design process, as shown in Fig. 21, 1) Assigning different permissions according to the different divisions of roles

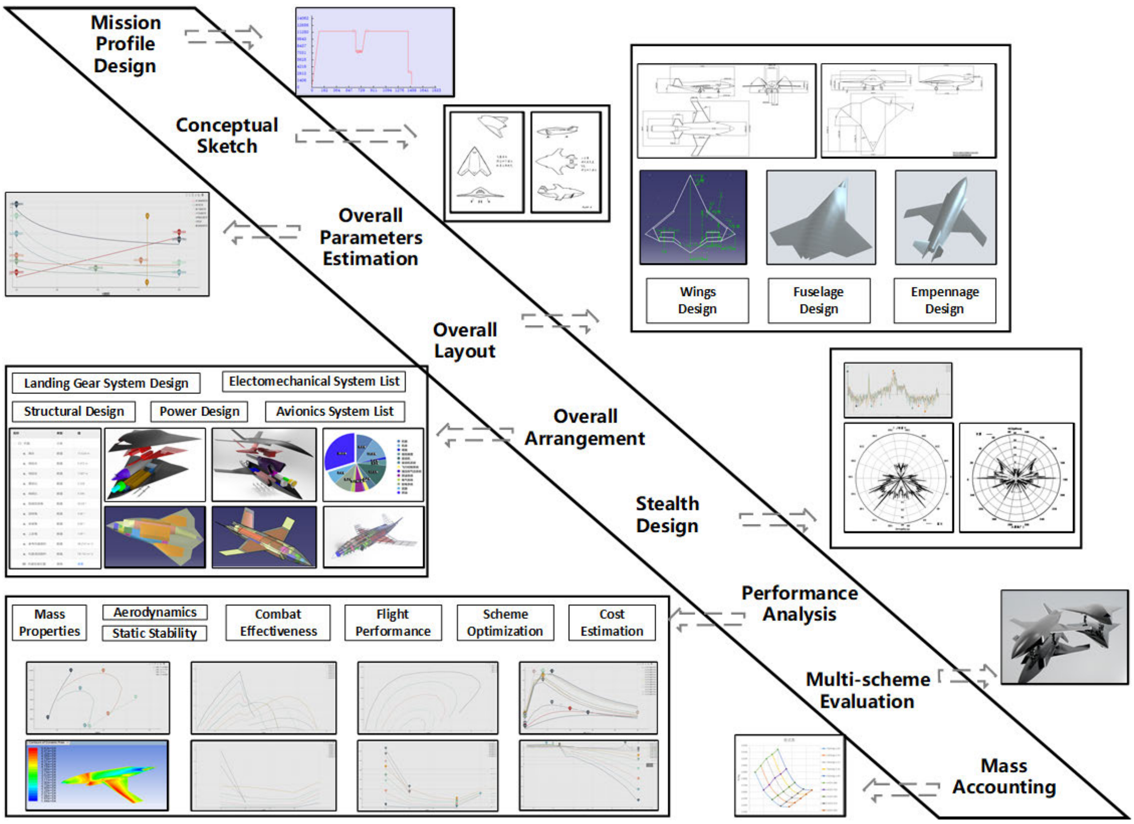
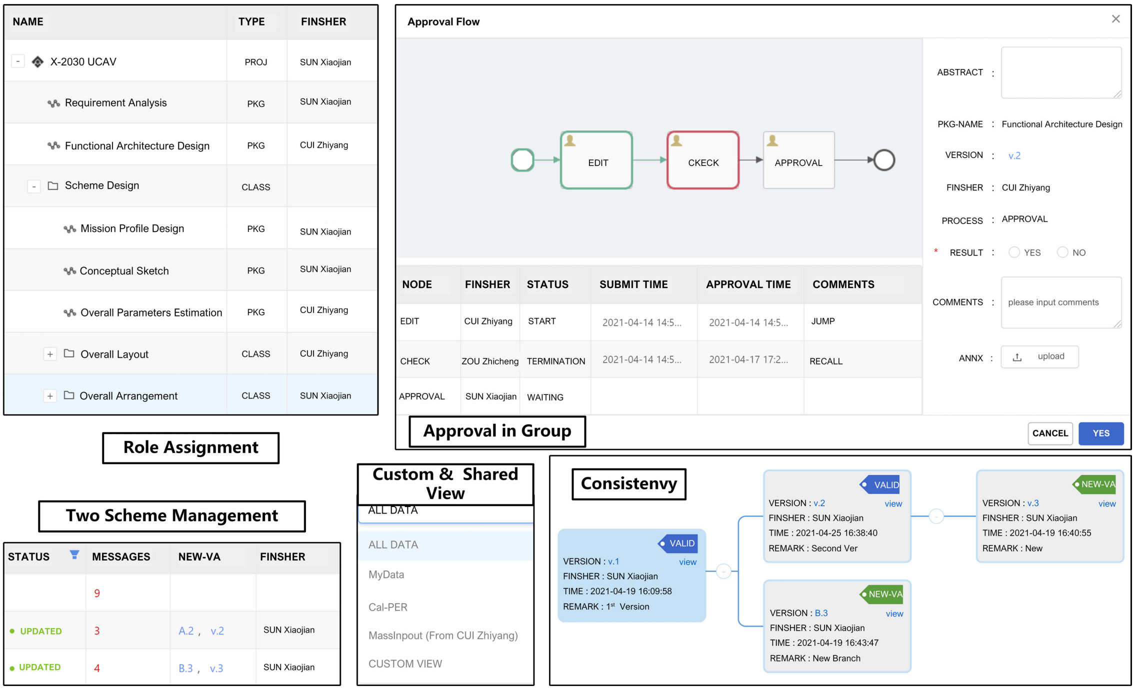


FIGURE 20. The design results of X-2030 UCAV.



Role Assignment

NAME	TYPE	FINSHER
X-2030 UCAV	PROJ	SUN Xiaojian
Requirement Analysis	PKG	SUN Xiaojian
Functional Architecture Design	PKG	CUI Zhiyang
Scheme Design	CLASS	
Mission Profile Design	PKG	SUN Xiaojian
Conceptual Sketch	PKG	SUN Xiaojian
Overall Parameters Estimation	PKG	CUI Zhiyang
Overall Layout	CLASS	CUI Zhiyang
Overall Arrangement	CLASS	SUN Xiaojian

Approval in Group

Approval Flow

NODE	FINSHER	STATUS	SUBMIT TIME	APPROVAL TIME	COMMENTS
EDIT	CUI Zhiyang	START	2021-04-14 14:5...	2021-04-14 14:5...	JUMP
CHECK	ZOU Zhicheng	TERMINATION	2021-04-14 14:5...	2021-04-17 17:2...	RECALL
APPROVAL	SUN Xiaojian	WAITING			

ABSTRACT :

PKG-NAME : Functional Architecture Design

VERSION : v.2

FINSHER : CUI Zhiyang

PROCESS : APPROVAL

* RESULT : ○ YES ○ NO

COMMENTS : please input comments

ANNX : upload

CANCEL YES

Two Scheme Management

STATUS	MESSAGES	NEW-VA	FINSHER
	9		
● UPDATED	3	A.2 , v.2	SUN Xiaojian
● UPDATED	4	B.3 , v.3	SUN Xiaojian

Custom & Shared View

ALL DATA

ALL DATA

MyData

Cal-PER

MassInpout (From CUI Zhiyang)

CUSTOM VIEW

Consistenvy

VALID view
VERSION : v.1
FINSHER : SUN Xiaojian
TIME : 2021-04-19 16:09:58
REMARK : 1st Version

VALID view
VERSION : v.2
FINSHER : SUN Xiaojian
TIME : 2021-04-25 16:38:40
REMARK : Second Ver

NEW-VA view
VERSION : B.3
FINSHER : SUN Xiaojian
TIME : 2021-04-19 16:43:47
REMARK : New Branch

NEW-VA view
VERSION : v.3
FINSHER : SUN Xiaojian
TIME : 2021-04-19 16:40:55
REMARK : New

FIGURE 21. X-2030 UCAV dynamic management.

to complete dynamic permission management. 2) Multiple approvals in the group to ensure credibility. 3) Using dynamic version management to standardize the consistency of versions across disciplines in the design of conceptual schemes. 4) Managing the design of two schemes to ensure that the model and data are not misaligned, reducing complexity.

5) Dynamic view management, customizing views within the group and sharing them to assist designers in making decisions and working with each other.

D. CASE STUDY CONCLUSION

The validation of X-2030 UCAV shows that the AST system can support the entire process of UCAV conceptual design. The AST system can capture, manage, and apply the models and data of UACV conceptual design and help manage the models and data in multi-person collaborative design. As conclusion, the validation of the X-2030 UCAV case study shows that the AST system achieves its construction objects mentioned in Section II.

1) Improving the models and data reusability: The X-2030 UCAV co-design model is repeatedly applied during the design process, which reduces the time for a single design.
2) Eliminating inconsistencies: Since the models and data are centralized in the AST system, designers can get the latest models and data from the AST system. This reduces inconsistencies in the multi-person collaboration process.
3) Providing accurate models and data. The models and data provided by the AST system are accurate models and data that have been tested and approved by the group.
4) Reducing complexity. The same design group can perform 36 iterations for X-2030 UCAV design without AST system and 45 iterations for X-2030 UCAV design with AST system in the same design cycle. The AST system can reduce the complexity and improve the efficiency by about 25%.
5) Enhancing management and assist in decision making. One of the reasons for the increased efficiency of the X-2030 UCAV project design is the AST system's ability to enhance management. Designing with the AST system also allows for quick access to models and data during the review phase through the dynamic view management, which assists in decision making.

The AST system has been proven to help design teams shorten the design cycle, improve design efficiency, and ensure the accuracy of design models and data compared to traditional co-design methods.

V. CONCLUSION

The construction of an AST system is the basis for implementing MBSE, which can be accessed to obtain models and data at any stage of the entire life cycle, helping engineers confirm whether the requirements are met. The AST can help engineers reduce their workload and focus on consistent and reliable models and data without worrying about the wrong model data generated by management confusion. At the same time, the AST can reduce the management complexity of managers, all project progress at a glance, through the AST to provide derivative tools, and managers can view the data of the system. There is no need to organize numerous meetings for reporting.

By defining the objectives and basic methods of building the AST system, this paper designed AST architecture and studied various building techniques of the AST system. Based on the successful building of the AST, with the help of the UCAV conceptual design, we confirm that the AST system can meet the requirements of the conceptual design stage and show the feasibility of building techniques and the integrity of the system.

The design, construction, and maintenance of the AST system are also the practice of system engineering and software engineering, which requires continuous exploration and practice. At present, there are still some technologies and functions that need to be studied.

1) Add element types to the AST system, study the models and data of various disciplines in the detailed design phase, sort out their expressions, and add element types not involved in the conceptual scheme design phase, such as electromechanical and avionics.
2) Research traceability techniques to help conduct rapid traceability of requirements and design solutions and form traceability relationship diagrams to help engineers intuitively understand the traceability situation.
3) Use the massive models and data in the AST system to construct knowledge graphs and assist system engineering evolution through knowledge engineering.

ACKNOWLEDGMENT

The authors thank to all the members of the DT-316 Team from the Aircraft Overall Design Course, in 2019, who designed the X-2030 UCAV, which provided the basis of case study.

REFERENCES

[1] S. R. Hirshorn, L. D. Voss, and L. K. Bromley, *NASA Systems Engineering Handbook*. Washington, DC, USA: NASA, 2017.

[2] P. A. Jansma and R. M. Jones, "Advancing the practice of systems engineering at JPL," in *Proc. IEEE Aerosp. Conf.*, Mar. 2006, p. 19, doi: 10.1109/AERO.2006.1656171.

[3] *INCOSE, Systems Engineering Vision 2020, International Council on Systems Engineering (INCOSE)*, document INCOSE-TP-2004-004-02, Technical Operations, Sep. 2007.

[4] *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed, INCOSE, San Diego, CA, USA, 2015.

[5] G. F. Dubos, D. P. Coren, A. Kerzhner, S. H. Chung, and J.-F. Castet, "Modeling of the flight system design in the early formulation of the Europa project," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–14, doi: 10.1109/AERO.2016.7500604.

[6] T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, and D. Wagner, "Update on the model based systems engineering on the Europa mission concept study," in *Proc. IEEE Aerosp. Conf.*, Mar. 2013, pp. 1–13, doi: 10.1109/AERO.2013.6496855.

[7] T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner, "Early formulation model-centric engineering on NASA's Europa mission concept study," in *Proc. INCOSE Int. Symp.*, vol. 22, no. 1, 2012, pp. 1695–1710, doi: 10.1002/j.2334-5837.2012.tb01431.x.

[8] J. Lu, J. Wang, D. Chen, J. Wang, and M. Torngren, "A service-oriented tool-chain for model-based systems engineering of aero-engines," *IEEE Access*, vol. 6, pp. 50443–50458, 2018, doi: 10.1109/ACCESS.2018.2868055.

[9] L. Li, N. L. Soskin, A. Jbara, M. Karpel, and D. Dori, "Model-based systems engineering for aircraft design with dynamic landing constraints using object-process methodology," *IEEE Access*, vol. 7, pp. 61494–61511, 2019, doi: 10.1109/ACCESS.2019.2915917.

[10] S. Gao, W. Cao, L. Fan, and J. Liu, "MBSE for satellite communication system architecting," *IEEE Access*, vol. 7, pp. 164051–164067, 2019, doi: 10.1109/ACCESS.2019.2952889.

[11] R. Krishnan and S. V. Bhada, "An integrated system design and safety framework for model-based safety analysis," *IEEE Access*, vol. 8, pp. 146483–146497, 2020, doi: 10.1109/ACCESS.2020.3015151.

[12] *Department of Defense Digital Engineering Strategy*, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Washington, DC, USA, Jun. 2018.

[13] B. Kruse and M. Blackburn, "Collaborating with OpenMBEE as an authoritative source of truth environment," *Proc. Comput. Sci.*, vol. 153, pp. 277–284, Jan. 2019, doi: 10.1016/j.procs.2019.05.080.

[14] *OpenMBEE Open Model Based Engineering Environment*. Accessed: Oct. 5, 2021. [Online]. Available: https://www.openmbee.org/index.html

[15] MBSE Wiki. *Mbse:Authoritative_Source_of_Truth*. Accessed: Jul. 14, 2020. [Online]. Available: http://www.omgwiki.org/MBSE/doku.php?id=mbse:authoritative_source_of_truth

[16] X. Sun, M. Luo, C. Zhang, R. Guan, and H. Liu, "Construction technology of authoritative source of truth for civil aircraft pre-research," *Acta Aeronautica et Astronautica Sinica*, vol. 42, no. 2, pp. 127–141, 2021.

[17] K.-B. Yue, M. Guerra, H. Wagner, J. S. Thamarai Selvan, K. Collector, V. Tang, M. Sikes, K. Sha, P. Kallempudi, S. Mardani, and K. Kasichainula, "Applying blockchain technology on model-based systems engineering," in *Proc. AIAA Scitech Forum*, Jan. 2021, pp. 11–21, doi: 10.2514/6.2021-0093.

[18] T. J. Bayer, L. A. Cooney, C. L. Delp, C. A. Dutenhoffer, R. D. Gostelow, M. D. Ingham, J. S. Jenkins, and B. S. Smith, "An operations concept for integrated model-centric engineering at JPL," in *Proc. IEEE Aerosp. Conf.*, Mar. 2010, pp. 1–14, doi: 10.1109/AERO.2010.5446799.

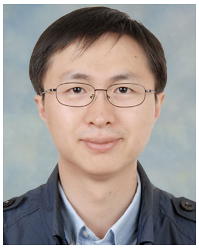
[19] *Effective Model-Based Systems Engineering*. New York, NY, USA: Springer, 2018.

XIAOJIAN SUN was born in Shaanxi, China, in 1995. He received the B.E. degree in aircraft design and engineering from Xi'an Jiaotong University, Xi'an, China, in 2016, and the M.E. degree in aeronautical and astronautical science and technology from the National University of Defense Technology, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree in aircraft design with the School of Aeronautic Science and Engineering, Beihang University. His research interests include aircraft conceptual design, model-based systems engineering, and the authoritative source of truth.

MINGQIANG LUO was born in Sichuan, China, in 1981. He received the B.E. and Ph.D. degrees in aircraft design from Beihang University, in 2004 and 2008, respectively. From 2008 to 2011, he was a Research Assistant with the Aircraft Advanced Design Technology Laboratory, Beihang University, where he has been a Lecturer with the School of Aeronautic Science and Engineering, since 2011, and an Associate Professor, since 2020. He is currently a main instructor of aircraft conceptual design which is a national quality curriculum. His main research interests include aircraft conceptual design, digital twin, model-based systems engineering, and big data.

ZHIYANG CUI was born in Gansu, China, in 1996. He received the B.E. degree in aircraft design and engineering from Beihang University, Beijing, China, in 2019, where he is currently pursuing the M.E. degree in aircraft design with the School of Aeronautic Science and Engineering. His research interests include aircraft conceptual design, model-based systems engineering, and computer-aided design technology.

ZHICHENG ZOU was born in Shandong, China, in 1998. He received the B.E. degree in aircraft design and engineering from Beihang University, Beijing, China, in 2021, where he is currently pursuing the Ph.D. degree in aircraft design with the School of Aeronautic Science and Engineering. His research interests include nonlinear aeroelasticity and overall design of aircraft.



HU LIU received the B.S. and Ph.D. degrees in aircraft design from Beihang University (formerly the Beijing University of Aeronautics and Astronautics), Beijing, China, in 2000 and 2004, respectively. He is currently the Head of the Aircraft Department and a Professor with Beihang University. He is a main instructor of aircraft conceptual design which is a national quality curriculum. His main research interests include aircraft conceptual design and training support technologies. He is a member of the American Institute of Aeronautics and Astronautics (AIAA) and the Chinese Society of Aeronautics and Astronautics (CSAA).