

# MBSE for Civil Aircraft Scaled Demonstrator Requirement Analysis and Architecting

ZHIYANG CUI<sup>1</sup>, MINGQIANG LUO<sup>1</sup>, CHI ZHANG<sup>2</sup>, YONGZHI HUA<sup>3</sup>,  
RAN XIE<sup>4</sup>, RUNXI YU<sup>2</sup>, AND CHEN HUANG<sup>2</sup>

<sup>1</sup>School of Aeronautic Science and Engineering, Beihang University, Beijing 100083, China

<sup>2</sup>Beijing Aircraft Technology Research Institute, Commercial Aircraft Corporation of China, Beijing 102211, China

<sup>3</sup>China Academy of Launch Vehicle Technology, Beijing 100076, China

<sup>4</sup>Yangzhou Collaborative Innovation Research Academy, Shenyang Aircraft Design and Research Institute, Yangzhou 225002, China

Corresponding author: Mingqiang Luo (luomingqiang\_buaa@163.com)

**ABSTRACT** A civil aircraft scaled demonstrator is an unmanned aerial vehicle (UAV) obtained by reducing the geometry of a civil aircraft by an equal scale and simplifying the airframe and airborne subsystems. Due to the low development cost and flight risk of scaled demonstrators, flight tests with demonstrators are an attractive way to assess the reliability and effectiveness of new configurations and/or technologies available for civil aircraft. Nevertheless, engineers still hope to reduce the development cycles and costs of demonstrators to evaluate their multiple civil aircraft design solutions through flight tests in a short period of time. Model-based systems engineering (MBSE) is a formalized requirement and architecture modeling method which is a useful tool in aircraft design. However, no existing MBSE framework has been found suitable for the development of demonstrators due to their unique features. In this paper, an MBSE approach for civil aircraft scaled demonstrator is proposed based on the MagicGrid methodology and the existing technical process for the demonstrator. This approach formalizes the requirement analysis and architecting for demonstrators via system modeling language (SysML). Meanwhile, the stakeholders of civil aircraft scaled demonstrators are identified, and a requirement ontology and modeling standard are established according to existing demonstrator development practice. Then, a case study is performed through a hybrid power demonstrator which carries a hydrogen fuel cell as the main power supply, a set of solar cells, and two lithium cells. The requirement traceability and verifiability are examined, and a logic simulation is executed for architecture, which shows the feasibility of applying the MBSE approach for the development of civil aircraft scaled demonstrators.


**INDEX TERMS** Scaled demonstrator, aircraft design, MBSE, MagicGrid.

## I. INTRODUCTION

Systems engineering (SE) is an interdisciplinary approach and means to enable the realization of successful systems [1]. Text-based SE focuses on documenting requirements in the early stages of product development and then proceeding with design synthesis, verification, and validation while considering the complete lifecycle of the system. In 2007, formalized modeling methods were introduced into SE [2]. These methods produce generic, unambiguous, and machine-readable models, endowing model-based systems engineering (MBSE) with efficiency and interoperability.

Aircraft design and spacecraft design, with high complexity and involves interdisciplinary approaches and advanced

technologies, are mainly based on system engineering [3], [4]. NASA has applied the MBSE method in several space missions, such as Europa Project [5]–[7], constructing a Single-Source-of-Truth for the early formulation of the mission concept. There are also applications of MBSE to aero-engines design [8], which proposes a service-oriented tool-chain with an emphasis on domain-specific views. Reference [9] presents an MBSE framework using an object-process methodology and incorporating the framework into a model-based design domain for landing gears with dynamic landing constraints. Reusable logical architecture models of satellite communication system is constructed in [10] after a comprehensive investigation of various methodologies. The models have detailed parametric constraints for model-based trade-off analysis. They hold the position that the models realized through the MBSE approach

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

are reusable and easily extendible to detailed system design and implementation throughout the product lifecycle.

A standardized and robust modeling language is considered a critical enabler for MBSE [11]. Systems Modeling Language (SysML) is a general-purpose graphical representation for the various disciplines of systems engineering. It was developed by the International Council on Systems Engineering (INCOSE) in cooperation with the Object Management Group (OMG) and is the de facto language of MBSE [1]. SysML consists of 9 different diagrams describing the requirements, structure, and behavior of the system and its components, supporting the specification, design, analysis, and verification of systems.

Another enabler for MBSE is the MBSE methodology. An MBSE methodology can be characterized as the collection of related processes, methods, and tools used to support the discipline of systems engineering [12]. Some of the MBSE methodologies used in the industry include Harmony-SE, OOSEM, Rup SE, and MagicGrid. The MagicGrid approach for MBSE by No Magic is a tool-independent modeling methodology and fully compatible with SysML. The approach includes the definition of problem domain with the Stakeholder Needs development process, solution domain with the Architecture Definition process, and implementation domain with the Design Definition process. The four pillars of the MagicGrid approach match the four aspects of the SysML: requirements, behavior, structure, and parameters. The approach has evolved by summarizing the experience of numerous MBSE adoption projects from various industry sectors, such as defense, automotive, aerospace, and health care [13].

A scaled demonstrator is a downscaled unmanned aerial vehicle (UAV) which is free-flown in the open atmosphere to obtain qualitative or quantitative information about a larger vehicle, a more complex system, or a technology of interest [14] and is very commonly used in aviation industry. Scaled demonstrators are sometimes referred to as downscaled UAVs or flying prototypes. Compared to a full-scale aircraft, the demonstrator has a lower design cost, shorter development cycle, and the ability to conduct flight tests in a real-world airspace environment with less flight safety risk [15]. Also, demonstrators are always produced in small batches, which is very different from commercial UAVs. Research as in [16] describes the design and flight tests of a joint wing scaled demonstrator, which is the application of demonstrators in the study of future airplane configurations. T-Flex is a 65kg scaled demonstrator designed to develop innovative technologies as aeroelastic wing and active flutter suppression [17].

As an innovative product with novel configurations or technologies, the development of scaled demonstrators needs a top-down requirement-driven approach such as MBSE. MBSE approach can realize the coupling and iteration of technical processes in the single software environment, enabling the rapid development of demonstrators. The relations constructed in the models can provide a subjective

approach to check the traceability and verifiability of the requirements. Moreover, the reuse of the models can shorten development cycle and facilitate future demonstrator projects. MBSE has been extensively utilized for requirement analysis and architecting of aerial vehicles in literature. However, there is a lack of consideration of the MBSE approach for civil aircraft scaled demonstrators which is a special type of aircraft with unique features. Research as in [18] implements a small unmanned aircraft system (SUAS) development environment capable of rapid prototyping. The low cost and use of commercial-off-the-shelf components for SUAS are similar to a civil aircraft demonstrator. However, the research lacks a process comparable to the requirement analysis of a demonstrator. An MBSE approach in [19] implements a certification-driven design for a more complex UAV with a T-tail configuration and a high aspect ratio of 19. The UAV is a full-scale aircraft meeting the airworthiness regulation for large aircraft (EASA CS-25 in this case). Its stakeholder needs, functional scenarios, and structures are quite different from civil aircraft scaled demonstrators.

In order to realize the benefits promised by MBSE in the development of civil aircraft scaled demonstrator, and thus further reduce the development cost and cycle, this paper proposes a formalized requirement analysis and architecting approach for the overall design of a demonstrator. The approach is based on MagicGrid methodology and enables a top-down requirement-driven design of civil aircraft demonstrators. The major steps of the approach are stakeholder needs capture, function analysis, requirement analysis and design synthesis, shown in Fig.3. Then, requirement analysis and architecting are performed for the LQ-H, a hybrid electric power demonstrator, and obtain a traceable, achievable, verifiable, and logically self-consistent result. The innovations of this paper are as follows.

- 1) Analyze the lifecycle process of civil aircraft scaled demonstrator, propose the MBSE approach for requirement analysis and architecting of demonstrator based on MagicGrid methodology, and apply the method to engineering practice.

- 2) For the requirement analysis of civil aircraft scaled demonstrator, define the ontology of requirements as well as the stakeholder needs, the stakeholder requirements, and the system requirements. Identify the stakeholders of demonstrators and propose a demonstrator requirement modeling standard.

- 3) Perform requirement analysis and architecting for an actual civil aircraft demonstrator and analyze the result to confirm that the MBSE approach proposed for demonstrators is effective.

The paper is organized as follows. Section II summarizes the technical process of the development of civil aircraft scaled demonstrator and describes the MBSE approach for requirement analysis and architecting of demonstrator based on the technical process. In section III, the stakeholders of demonstrators are identified, and the results of LQ-H's stakeholder needs are given. Then in section IV, the function

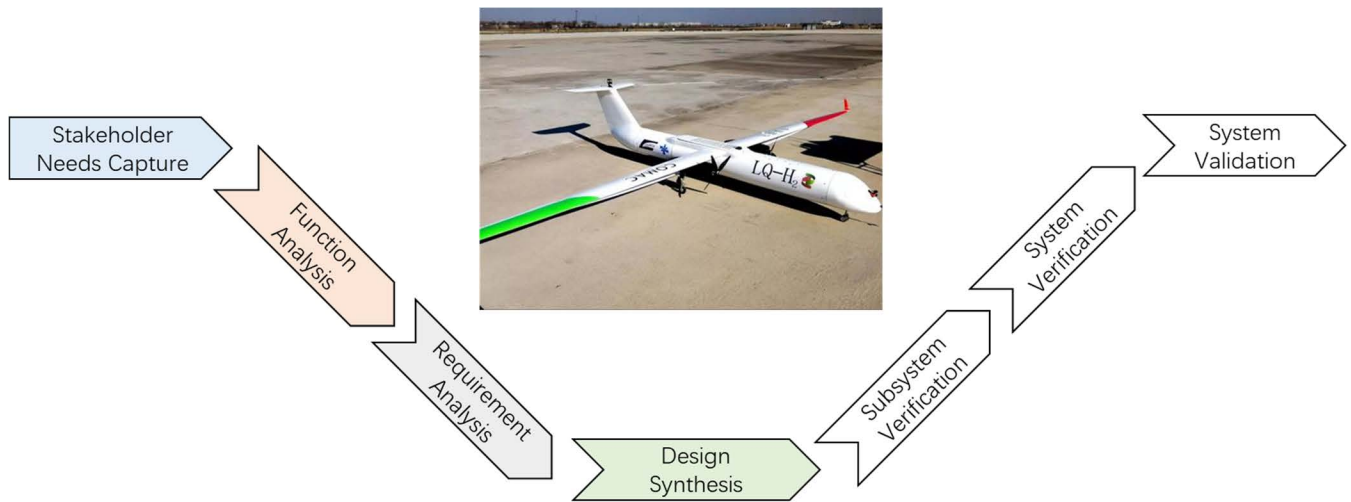


FIGURE 1. Technical process for demonstrator.

analysis of the demonstrator by means of decomposition is performed, and the architecting of the problem domain of LQ-H is completed. Section V illustrates the requirement ontology and requirement modeling standard for demonstrators and performs requirement analysis of LQ-H. In Section VI, the design synthesis completes the solution domain architecting of the demonstrator based on its requirements. Finally, Section VII analyzes the modeling result of requirement and architecture, while Section VIII concludes with benefits and future work.

## II. MBSE APPROACH FOR CIVIL AIRCRAFT SCALED DEMONSTRATOR

### A. TECHNICAL PROCESS FOR DEMONSTRATOR

Beijing Aircraft Technology Research Institute (BATRI) has formed a mature technical process for the development of civil aircraft scaled demonstrators in practice, as shown in Fig.1. The technical process is a non-model SE approach based on the SE technical process for commercial transportation aircrafts as in [20].

#### 1) STAKEHOLDER NEEDS CAPTURE

The purpose of stakeholder needs capture is to clarify which configuration or technologies the demonstrator needs to test and what demands the stakeholders will have for the demonstrator to complete that test. In BATRI demonstrator development practice, the top stakeholder needs document is called Voice of the Customers (VoC). VoC can usually be obtained directly from various stakeholders through questionnaires, interviews, workshops, etc.

Needs are defined differently from requirements in this paper. Needs are usually qualitative and are not even determined to be achievable at the initial stage of requirement analysis. In contrast, requirements are descriptions of the performance or design constraints that a product must meet. Requirements must be achievable and verifiable.

#### 2) FUNCTION ANALYSIS

In function analysis, the functions of the demonstrator are analyzed based on stakeholder needs and experience from other demonstrator projects to obtain a complete description of the expected functions of the demonstrator. The description includes a list of the functions of the demonstrator, the architecture implementing the functions, and the interfaces between the elements of the architecture.

#### 3) REQUIREMENT ANALYSIS

After completing the stakeholder needs capture and function analysis, the requirements analysis process is to derive the stakeholder needs into the design requirements of the demonstrator based on the results of the above two technical processes. In the practice of system engineering, the system is decomposed into layers of hierarchy. Each layer has corresponding requirements, and there are tracing and derivative relations between requirements of different layers. In BATRI demonstrator projects, requirements of civil aircraft scaled demonstrators are divided into four layers: stakeholder, aircraft (i.e., System of Interest, SoI), subsystem, and component, shown as Fig.2.

At the stakeholder layer, VoC is the topmost stakeholder needs described above. Design Requirements and Objectives (DRO) is stakeholder requirements derived from VoC, which is required to be an achievable and verifiable description of the system. In the requirements analysis process, engineers develop DRO based on system functional architecture, past project experience, and discipline knowledge. DRO is the bridge between stakeholder needs and system requirements. During the analysis of DRO, the requirement engineers maintain close communication with stakeholders and subsystem experts and conduct multiple iterations between function analysis and DRO. The resulting DRO should accurately reflect the stakeholder needs for the SoI and provide a

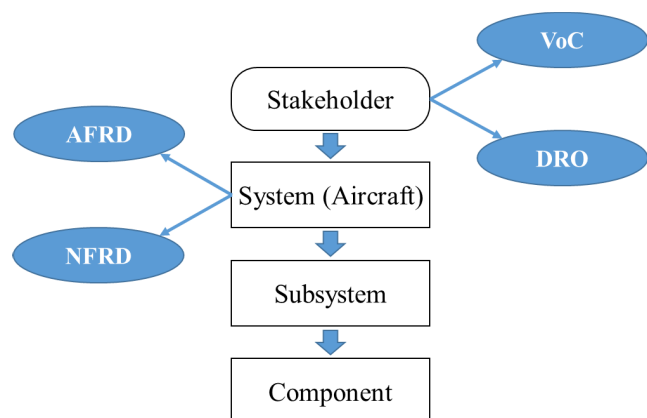


FIGURE 2. Layers of hierarchy of a demonstrator.

well-defined and professional design objective for the design synthesis process.

Aircraft requirements are derived from DRO and can be divided into Aircraft Functional Requirements Document (AFRD) and Non-Functional Requirements Document (NFRD) according to whether they are related to the functions of the demonstrator. It is necessary to distinguish between AFRD and NFRD because these two types of requirements have different approaches for verification. Functional requirements can be verified by the behavior or interfaces of the system. Tests of the relevant properties can verify whether the requirements are met once the system has been built. Non-functional requirements are irrelevant with behavior models. Some of them can be verified by the parameters of the system (e.g., weight, price), while others, like maintenance and manufacturability, may require some specialized test to verify.

Subsystem requirements and component requirements are derived from aircraft requirements and should also be distinguished according to whether they are functional or not. In practice, subsystems and components of a demonstrator are usually selected from off-the-shelf products or provided by stakeholders. Therefore, the design synthesis can be performed once the system-level requirements analysis is completed.

#### 4) DESIGN SYNTHESIS

For civil aircraft scaled demonstrator, the task of the design synthesis is to select the appropriate flight vehicle and airborne systems based on the requirements and the architecture. Since the demonstrator is to perform stakeholder-specified flight tests, some of the subsystems may be test equipment provided by stakeholders. If an incompatibility is found between the subsystem provided by the stakeholder and the design requirements of the demonstrator, the requirement analysis process should be checked and modified, iterating until the incompatibility disappears.

#### 5) VERIFICATION AND VALIDATION

Subsystems selected in the design synthesis process is verified if they meet the requirements. If the verification passes,

they will be integrated to SoI. Then system-level tests will be performed to verify that the demonstrator meets all requirements in AFRD and NFRD. Finally, the demonstrator will be put into flight test and be validated to meet the stakeholder needs.

### B. MAGICGRID FOR CIVIL AIRCRAFT SCALED DEMONSTRATOR

The proposed MBSE approach for civil aircraft scaled demonstrator based on the MagicGrid methodology and the technical process is shown in Fig.3. This approach is compatible with the abovementioned non-model SE process. The compatibility is important since it determines whether the MBSE approach can be applied in demonstrator development smoothly and permanently. The approach allows for stakeholder needs capture, function analysis, requirements analysis, and design synthesis of a demonstrator. Verification and validation involves complex and multidisciplinary tests which are out of the scope of MagicGrid methodology and is not included in the proposed approach. The correspondence between the demonstrator development technical process and the blocks in MagicGrid is identified by colors in the figures. The modeling workflow of the approach and the relationship between the models are also indicated in the figure. The MBSE approach for demonstrators can be divided into four steps:

The first step is to identify the stakeholders of the demonstrator and gather their needs from each stakeholder to form a VoC model.

In the second step, the functional scenario analysis of the demonstrator is carried out according to the stakeholder needs, and the construction of the behavior and structure models of the black box is performed. White box behavior and structure modeling are then performed according to the decomposition of the system functions of the demonstrator.

In the third step, VoC is derived into DRO, and Measurements of Effectiveness (MoEs) are developed. The problem domain models will be modified in multiple iterations. The researchers will need to maintain communication with stakeholders and subsystem experts to ensure that the requirements accurately reflect stakeholder needs and are achievable and verifiable. The solution domain system requirements modeling is then initially performed based on DRO. The system requirements are classified into AFRD and NFRD based on whether the requirements are related to the system functions.

In the final step, the solution domain system structure and behavior are modeled according to the system requirements, while the system requirements are iterated in this process. Maintain communication with subsystem experts to ensure that the system requirements are reasonable. After determining the system-level models, the products that meet the design requirements are selected from the optional subsystems, and their behavior and structure should be modeled.

Unlike the original MagicGrid method, the requirements of black-box in the MBSE approach for demonstrators is divided into two models: stakeholder needs and stakeholder

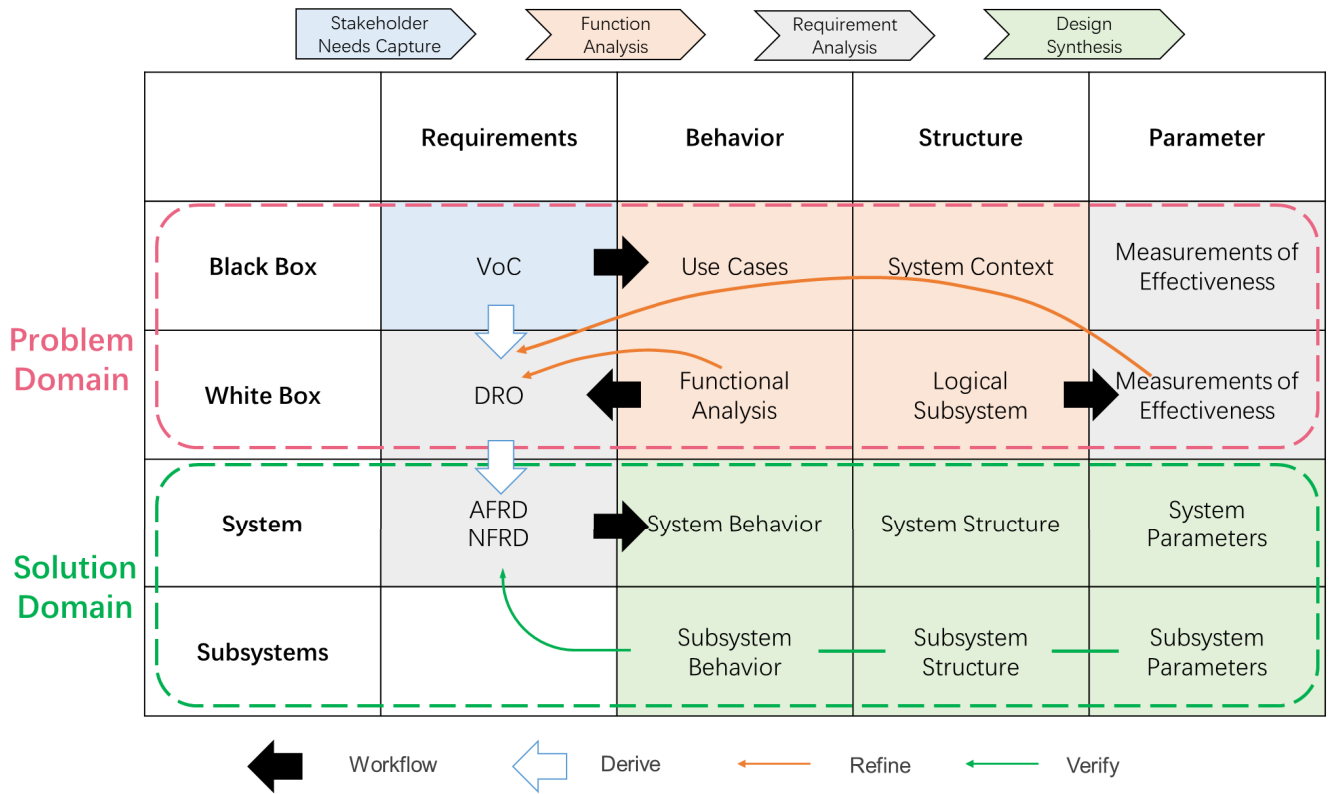


FIGURE 3. MagicGrid for civil aircraft scaled demonstrator.

requirements. The adjustment is to make the MBSE approach compatible with non-model technical process in the previous section. With the experience from our past demonstrator projects, we have found that the system requirements of demonstrators are quite complex and the derivation from stakeholder needs directly to system requirements might be insufficient of basis. With the stakeholder requirements added in the requirement analysis, it is helpful to derive stakeholder needs to system requirements based on the refinement of function analysis.

### III. STAKEHOLDER NEEDS CAPTURE

#### A. STAKEHOLDERS IDENTIFICATION FOR CIVIL AIRCRAFT SCALED DEMONSTRATOR

A stakeholder is any entity (individual or organization) with a legitimate interest in the system [13]. Stakeholders that generally need to be considered in a civil aircraft requirement analysis include Customer, Operational, Market, Training/Service, Maintenance, OEM (Original Equipment Manufacturer)/Supplier, and Regulatory [21]. In this paper, stakeholders of demonstrators are identified likewise.

1) Customer: For civil aircraft, Customer is usually defined as airlines, pilots, passengers, etc. For scaled demonstrators are defined as the test user. The test users are the operator in the flight tests of demonstrators and are also responsible for the preparation and aftercare of the test (e.g., transport and recovery of demonstrators).

2) Operational: For civil aircraft, Operational includes air traffic control, airports, etc. For demonstrators defined as airport and environment, this stakeholder is mainly concerned with the requirements of airports and airspace conditions on the characteristics of demonstrators.

3) Market: Market includes the requirements of market competitiveness. For a demonstrator, it is defined as collaborating users. The co-users can be considered as the procurer of demonstrators who complete the purpose of demonstrating specific technologies or configurations. To achieve this, the co-users usually set requirements for demonstrators' payload, internal space, and flight performance. And sometimes, they might also require that the demonstrator carries certain equipment or adopt a certain configuration.

4) Training/Service: Flight crews, cabin crews, ground service, passengers, etc., for civil aircraft. Demonstrators are simpler systems operated by professional staff from the development team or test users and usually do not require additional training or ground service. At the same time, demonstrators are customized for the co-users and are not intended for non-specific users/passengers. Therefore, this stakeholder is not considered for demonstrators.

5) Maintenance: For civil aircraft, including repairers, third-party repair companies, etc. Demonstrators are generally maintained by the development team or test users, without dedicated maintenance staff or third-party maintenance company.

TABLE 1. LQ-H voice of the customers.

ID	Name	Text
LQH-VoC-0001	Co-Users Requirements	
LQH-VoC-0001.1	Payload Weight	LQ-H can carry a payload of more than 8kg.
LQH-VoC-0001.2	Power Sources	LQ-H should have hybrid-electric power.
LQH-VoC-0001.3	Payload Volume	LQ-H should have space more than TBDcm*TBDcm*TBDcm for payload.
LQH-VoC-0001.4	Configuration	LQ-H should be truss-braced-wing (TBW) with two propellers.
LQH-VoC-0001.5	Payload Control	LQ-H should have the ability to active payload control.
LQH-VoC-0001.6	Means of Take-off and Landing	LQ-H should have wheeled landing gears.
LQH-VoC-0001.7	Flight Speed	The maximum speed of LQ-H should be more than 100km/h.
LQH-VoC-0001.8	Flight Duration	The flight duration of LQ-H should be more than 12h.
LQH-VoC-0001.9	Total Weight	The total weight of LQ-H should be less than 40kg.
LQH-VoC-0001.10	Ceiling	The ceiling of LQ-H should be more than 4000m.
LQH-VoC-0002	Development Team Requirements	
LQH-VoC-0002.1	Development Procedure	LQ-H should be developed under the MBSE procedure.
LQH-VoC-0002.2	Development Period	LQ-H should be developed within 4 months.
LQH-VoC-0002.3	Development Resource	
LQH-VoC-0002.3.1	Work Hour	LQ-H should be developed within 4000 work hours.
LQH-VoC-0002.3.2	Funding	The development of LQ-H should cost less than 13,000USD.
LQH-VoC-0002.4	Durability	LQ-H can operate test missions more than 200 times.
LQH-VoC-0002.5	Storage	LQ-H should be eligible for a specific storage site.
LQH-VoC-0003	Test Users Requirements	
LQH-VoC-0003.1	Transport	LQ-H can be transported in minivans.
LQH-VoC-0003.2	Flight Mission	LQ-H can fly in a specific route and maneuver.
LQH-VoC-0003.3	Flight Data	LQ-H can collect specific flight data.
LQH-VoC-0003.4	Pilot	LQ-H can be operated by a pilot of 2nd class or lower.
LQH-VoC-0003.5	Flight Safety	LQ-H can give an alert when it flies out of a certain area.
LQH-VoC-0004	Airport and Environment Requirements	
LQH-VoC-0004.1	Sidewind Speed	LQ-H can take-off and land in sidewind of more than 5m/s.
LQH-VoC-0004.2	Airport	LQ-H can satisfy the taking-off and lading requirements of a specific airport.
LQH-VoC-0004.3	Airspace	LQ-H can satisfy the requirements of operating in isolated airspace.
LQH-VoC-0004.4	Runway	LQ-H can take-off and land on a runway shorter than 400m.
LQH-VoC-0004.5	Wind	LQ-H can take-off and land in a wind of faster than 3m/s.
LQH-VoC-0004.6	Headwinds	LQ-H can take-off and land in a headwind of faster than 10m/s.
LQH-VoC-0004.7	Operating Temperature	
LQH-VoC-0004.7.1	Lowest Operating Temperature	The lowest operating temperature of LQ-H should be lower than 0°C.
LQH-VoC-0004.7.2	Highest Operating Temperature	The highest operating temperature of LQ-H should be higher than 40°C.

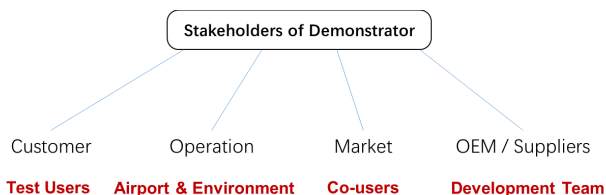


FIGURE 4. Stakeholders of civil aircraft scaled demonstrators.

6) OEM/Supplier: OEM/Supplier includes requirements from OEM strategy, policy, supplier cooperation, etc. For demonstrators, it can be defined as the development team whose main concerns are funding and work hours.

7) Regulatory: airworthiness regulatory (e.g., FAA, EASA), circular advisory requirements, and other applicable standards. Demonstrators usually do not require airworthiness certification, so Regulatory would not be considered a stakeholder.

The final results of identified stakeholders are co-users, test users, airport and environment, and development team, as shown in Fig.4.

**B. LQ-H VOICE OF THE CUSTOMERS**

The study case in this paper, LQ-H, is a hybrid electric power demonstrator developed by BATRI. The goal of the

demonstrator is to carry a hybrid power system consisting of hydrogen fuel cells, lithium batteries, and solar cells and test the performance of the fuel cells by flying in a natural airspace environment.

Based on the results of stakeholder identification, stakeholder representatives of LQ-H were convened to conduct a workshop and collect their needs. The stakeholder needs collected are modeled as VoC, shown in Table 1.

**IV. FUNCTION ANALYSIS**

**A. FUNCTIONAL SCENARIO ANALYSIS**

Functions are the capabilities that customers or potential users expect the product to possess. Functional scenario analysis is the basis for system behavior modeling. The functional scenarios are also divided into aircraft functional scenarios, subsystem functional scenarios, and component functional scenarios, corresponding to the system layers that implement the functions.

In the functional scenario analysis of demonstrators, the top-level use cases in the complete demonstrator operation workflow are first defined in chronological order: pre-flight inspection, preparation for take-off, mission operation, and post-flight inspection. The top-level use cases are then decomposed into activities with reference to the design

**TABLE 2. Functional scenario analysis for preparation for take-off.**

Use case	Aircraft Functional Scenario Activity	Subsystem Functional Scenario Activity
Preparation for Take-off	Ground Station Standby	
	Flight Control Communication On	Flight Control System Power on Flight Control System Activation Communication Established Communication Check
	Landing Site and Route Setting	
	Control Command Transmission Check	
	Control Command Unit Check	Alternate Mode Switching Alternate Mode Execution Main Control Mode Switching Main Control Mode Execution FPV Control Mode Switching FPV Control Mode Execution Automatic Check Automatic Check Report to Ground Station Automatic Check Report to Console

requirements and security specifications. Each activity has its corresponding execution system. If the execution system of the activity is the demonstrator, the activities can be continued to be decomposed into activities executed by subsystems.

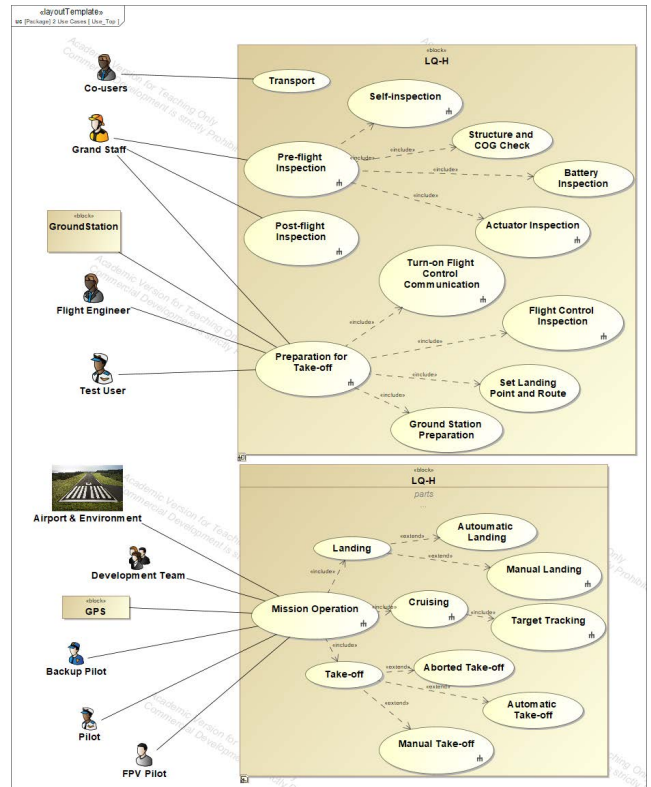
Table 2 shows the results of the functional scenario decomposition for LQ-H's preparation for take-off. According to the demonstrator operation specifications developed during the past demonstrator projects, the use case was divided into five aircraft functional scenario activities: ground station preparation, flight control communication turn-on, landing point and route setting, control command transmission check, and control command unit check. Among these activities, flight control communication turn-on and control command unit check are the activities performed by the demonstrator. Thus, these activities should be decomposed into subsystem functional scenario activities. Although there is no subsystem specified to perform a particular activity in the decomposition, it is necessary to consider the actual possible situation of the subsystems to avoid unreasonable subsystem functional scenario activities.

**B. USE CASE**

The use case diagram of the demonstrator is drawn on the basis of the functional scenario analysis above. In the initial modeling of the use case, it may not be possible to specify the entities that execute specific use cases. The use case can be completed after system context modeling. The final use case model of LQ-H is shown in Fig.5.

**C. SYSTEM CONTEXT**

System context is an external view of the demonstrator and includes elements that are not part of the system but are related to the system activity. In system context modeling,



**FIGURE 5. The use case diagram of LQ-H.**

an internal block diagram (IBD) is used to describe the system of systems (SoS) and its interfaces. When modeling system context, the functional scenario of the demonstrator should be considered. If activities that are not decomposed in the aircraft functional scenario cannot find an executor in the system context, it indicates that there are missing elements in the SoS model. The system context of LQ-H is shown in Fig.6. The experience of previous demonstrator projects is referred to the modeling.

**D. FUNCTIONAL ANALYSIS**

As the system activities and subsystem activities have been analyzed in the functional scenario, the main work of functional analysis modeling is to design the workflow and logic of the activities in activity diagrams. After the initial stage of functional analysis, the results are used as the basis for logical subsystems. Then with the results of the logical subsystem, the swim lanes can be created in the activity diagram, and the performers of each activity can be assigned. Fig.7 shows the final results of the LQ-H control command unit check activity.

**E. LOGICAL SUBSYSTEM**

The logical subsystem model of demonstrators is expressed through IBD, in which the subsystems of the demonstrator and the interfaces between them should be defined. The results of the logical subsystem should ensure that each activity in the functional analysis is performed by the corresponding subsystem, that there is a corresponding interface between

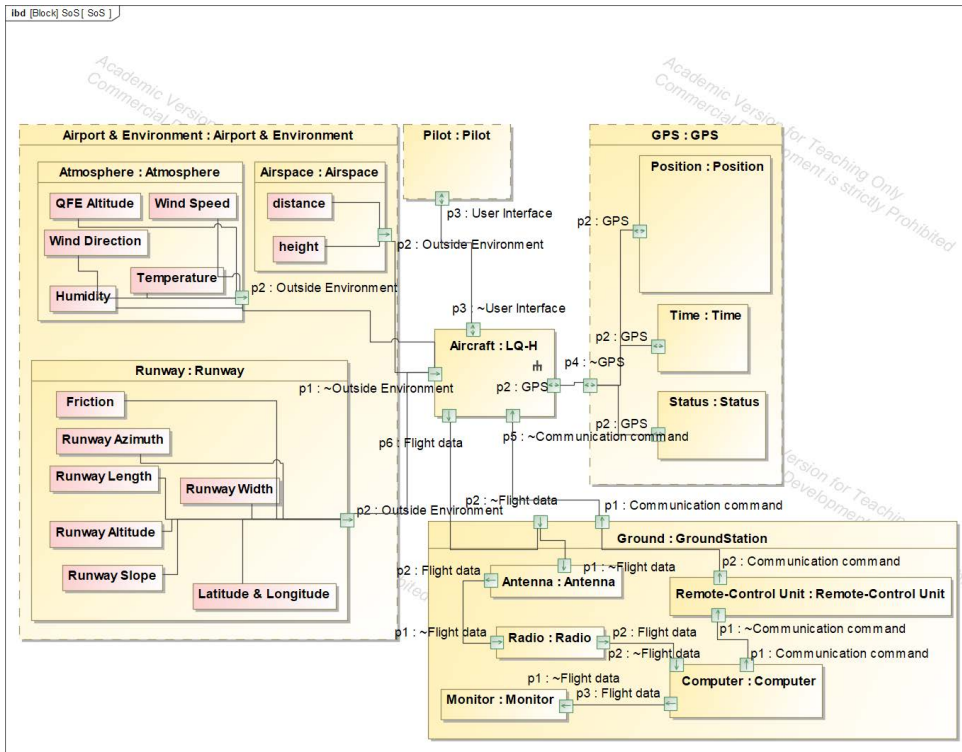


FIGURE 6. System context of LQ-H.

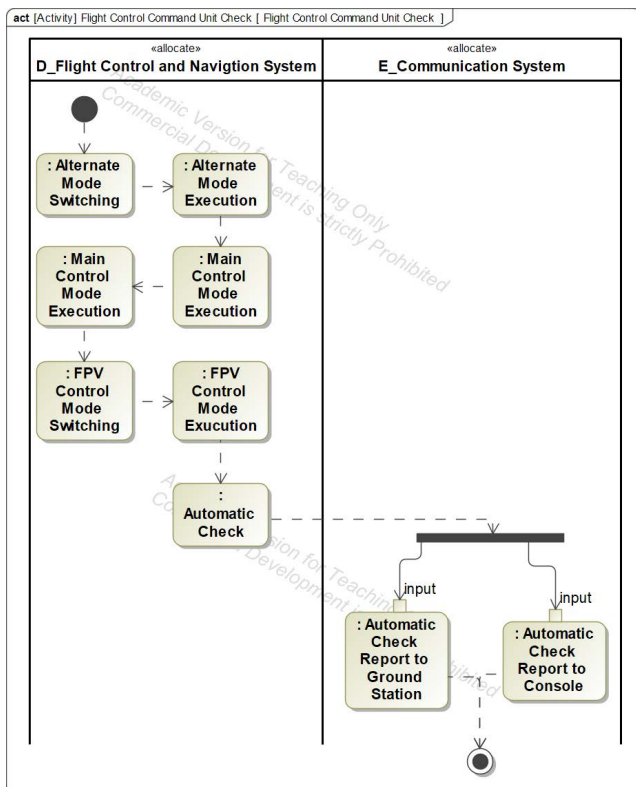


FIGURE 7. Activity diagram of LQ-H control command unit check.

the flows of the activities, and that the interface between the subsystem and the external system is consistent with that in the system context.

The logical subsystems of the LQ-H are shown in Fig.8. The demonstrator contains seven logical subsystems: airframe, flight control and navigation system, communication system, power system, data recording system, payload, and landing gear. The airframe and landing gear can sometimes be considered as one subsystem. However, this demonstrator uses a retractable landing gear with a complex mechanical structure, and therefore it is necessary to analyze the landing gear as a separate subsystem. Payload is defined as an unspecified subsystem that varies considerably depending on the stakeholder needs in different demonstrator projects. For LQ-H, the payload is a motorized gimbal and a camera mounted on it.

The above-mentioned tasks in the function analysis of demonstrators are tightly coupled. To ensure that the resulting behavior and structure model of the demonstrator is reasonable, it is important to maintain communication with stakeholders and subsystem experts during the actual modeling effort and iterate the tasks with experience from other demonstrator projects.

## V. REQUIREMENT ANALYSIS

### A. REQUIREMENT ONTOLOGY AND MODELING STANDARD

In the previous section, we have defined the requirement models for demonstrators. Before presenting the specifics of the requirement analysis for LQ-H, we first present the requirement ontology and requirement modeling standard that BATRI typically uses in its demonstrator projects.



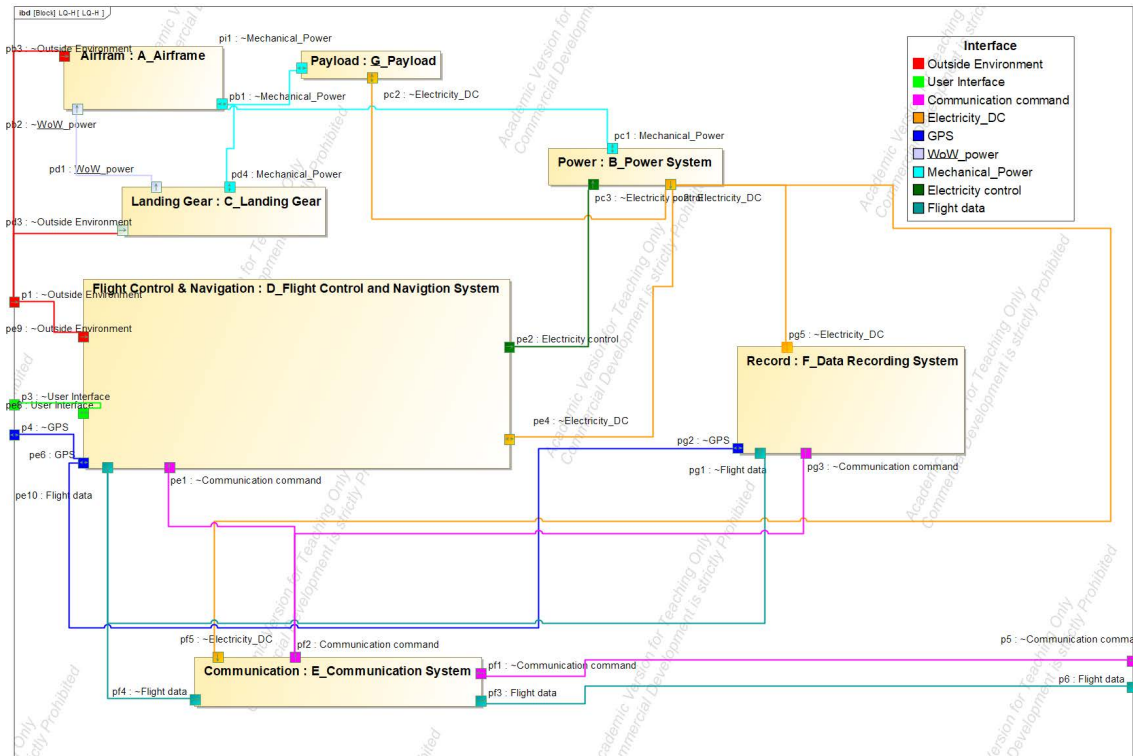


FIGURE 8. Logical subsystem of LQ-H.

1) DEMONSTRATOR REQUIREMENT ONTOLOGY

According to the convention of requirement descriptions in BATRI projects, the three elements of the requirement ontology are defined as ID, Name, and Text.

a: ID

The requirement ID is used to identify the requirement layer of hierarchy to which a requirement belongs and to distinguish the parent-child relationship of the requirement item. The ID naming format is “AAA-BBB-CCCC”, where “AAA” is the project name such as “LQH”; “BBB” is the name of the requirement model to which the requirement item belongs, such as “VOC”, “DRO”, etc.; “XXXX” is a numeric code, usually “0001”, “0002”, etc. If a requirement has sub-requirements, use period marks to distinguish the layer of sub-requirements, such as “0001.1”, “0001.1.2”.

b: NAME

The requirement name is a concise description of the overall content of the requirement, such as “data acquisition” or “landing and takeoff performance”.

c: TEXT

The requirement text is used to describe the specific content of the requirements. The main task of modeling requirements is to standardize the statement of requirement text to express the content of the requirements. A preferred requirement statement pattern is like “Object + Constraint + Value”, e.g., “The take-off taxi distance of LQ-H should be less

than 400m.” The requirement text is not compulsory in every requirement item, especially for the parent item which needs several child items to state its content completely. In Table 1, the VoC requirements for LQ-H are defined according to the above requirement ontology.

2) DEMONSTRATOR REQUIREMENT MODELING STANDARD

In order to ensure that the requirement analysis produces reasonable results, the demonstrator requirement modeling standard is summarized based on the experience of the past projects of BATRI. Regular reviews should be organized during the requirements modeling process to ensure that the requirement models conform to the standard.

a: UNAMBIGUITY

Unambiguity means that the understanding and interpretation of each requirement are unique to anyone who reads it. And for every requirement, it should be clear what work is to be done to meet that requirement, who exactly is to do that work, and which system (or systems) applies to that requirement.

b: CONSISTENCY

The same terminology should be used in the different requirements to describe the same system or concept.

c: BLACK-BOX

Requirements should describe the expected behavioral performances and characteristics of the system they specify and should not express the specific means of implementing those

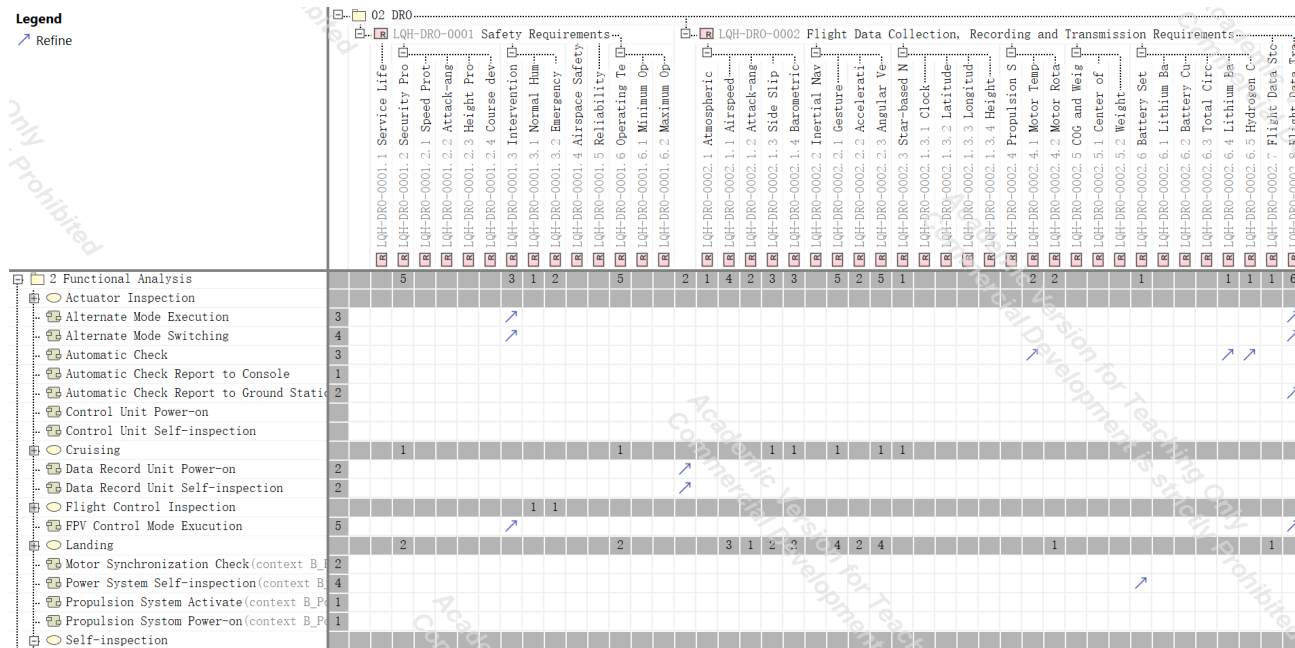


FIGURE 9. Part of refine matrix of functions and MoEs to DRO.

performances and characteristics unless the stakeholders have explicit demands on the implementation.

d: VERIFIABILITY

Each requirement must be verified to some extent through a standard method (e.g., test, analysis). For example, a customer’s expectation that an aircraft should have a range as long as possible is a legitimate request that cannot be verified. For such requirements, a trade-off analysis is required to determine the maximum range that can be verified. Ideally, each requirement can be verified by a separate test. If a requirement requires multiple tests to verify, the requirement should be divided into multiple requirements. If a single test can verify multiple requirements, the use of this test depends on whether the requirements can be categorized or merged.

e: ACHIEVABILITY

When defining the requirements, engineers with sufficient expertise should determine whether the requirements are achievable or not. To ensure that requirements are achievable, engineers responsible for the lower layer systems can be involved in the definition or reviews of requirements. The achievability of requirements should be assessed in terms of whether the metrics in the requirements can be met and whether the cost of meeting the metrics is affordable.

f: TRACEABILITY

Each requirement can be traced back to its source. Each requirement should be traced back layer-by-layer to a stakeholder need. And each stakeholder need must also be traced back to the corresponding stakeholder. Similarly, each function must be traced back to a specific requirement,

TABLE 3. Parent items of LQ-H DRO.

ID	Name
LQH-DR0-0001	Safety Requirements
LQH-DR0-0002	Flight Data Collection, Recording and Transmission Requirements
LQH-DR0-0003	Flight Performance Requirements
LQH-DR0-0004	Power Unit Requirements
LQH-DR0-0005	System Function Requirements
LQH-DR0-0006	Airframe Requirements
LQH-DR0-0007	Manufacture Requirements
LQH-DR0-0008	Maintenance Requirements
LQH-DR0-0009	Cost and Funding Requirements
LQH-DR0-0010	User Requirements

establishing a link between each requirement and the subsequent design, implementation, and testing.

B. DESIGN REQUIREMENTS AND OBJECTIVES

Stakeholders develop DRO from the results of functional analysis and logical subsystem. This step aims to complement the VoC by transforming some qualitative needs into measurable and verifiable requirements. Each white-box function should refine certain DRO items, thus ensuring that the DRO constrains all of the system’s functions. The complete DRO of LQ-H has 125 items. Due to the space limit, Table 3 shows the 10 parent items.

A refine matrix of functions and MoEs to DRO can be established during the requirement analysis process, as shown in Fig.9. The matrix can be used to check if all functions

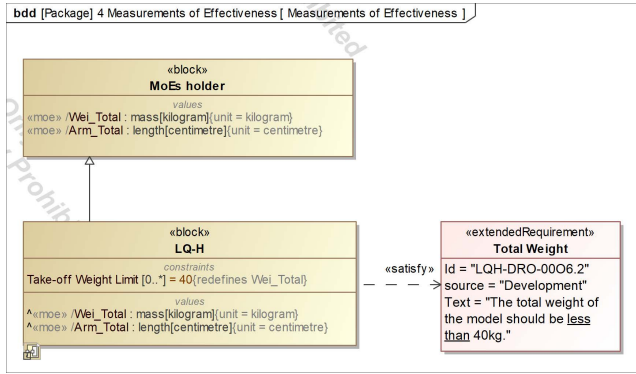


FIGURE 10. LQ-H total weight as MoEs.

and MoEs refine the DRO. And if not, the stakeholders should be advised to complete DRO items. Some of the DRO requirements cannot be refined by functions or MoEs. These requirements are usually non-functional and non-parameter, which should be treated carefully, e.g., involving subsystem experts in reviews to assess the achievability and verifiability of these requirements.

C. SYSTEM REQUIREMENT

The modeling process of system requirement is mainly to derive DRO into AFRD and NFRD, the definitions of which are presented in Section II. The specific tasks of the process are twofold.

The first is to categorize each DRO requirement in two models based on the previous refine matrix. The requirements that are refined by the functions are assigned to AFRD, and requirements that are not refined by the function assigned to NFRD. One of the reasons for this process is that functional

TABLE 4. Parent items of LQ-H AFRN and NFRD.

ID	Name
LQH-AFRD-0001	LQ-H Preparation for Take-off
LQH-AFRD-0002	Power System
LQH-AFRD-0003	Installation Interface and Functional Interface
LQH-AFRD-0004	Data Collection and Storage
LQH-AFRD-0005	Ground Taxiing
LQH-AFRD-0006	Data Transmission
LQH-AFRD-0007	Flight Status and Functional Systems Control
LQH-AFRD-0008	Flight Mode Switching
LQH-AFRD-0009	Landing and Recovery Requirements
LQH-NFRD-0001	Take-off and Landing Runway Requirements
LQH-NFRD-0002	Airframe Requirements
LQH-NFRD-0003	Cost and Funding Requirements
LQH-NFRD-0004	Maintenance Requirements
LQH-NFRD-0005	Reliability
LQH-NFRD-0006	Weight Limit
LQH-NFRD-0007	Manufacture Requirements
LQH-NFRD-0008	Preparation for Take-off Requirements

and non-functional requirements are associated with different models, resulting in different ways of verification. Another reason is that some of the non-functional requirements cannot be linked to other elements in the model and require additional attention during the requirements analysis.

The second is to adjust the organization and statement of the requirement items for the verifiability demand in the modeling standard, including quantifying non-quantified metrics in DRO, converting some hard-to-measure metrics into easy ones, and adjusting the organization of requirements so that the metrics verified by one single test is in the

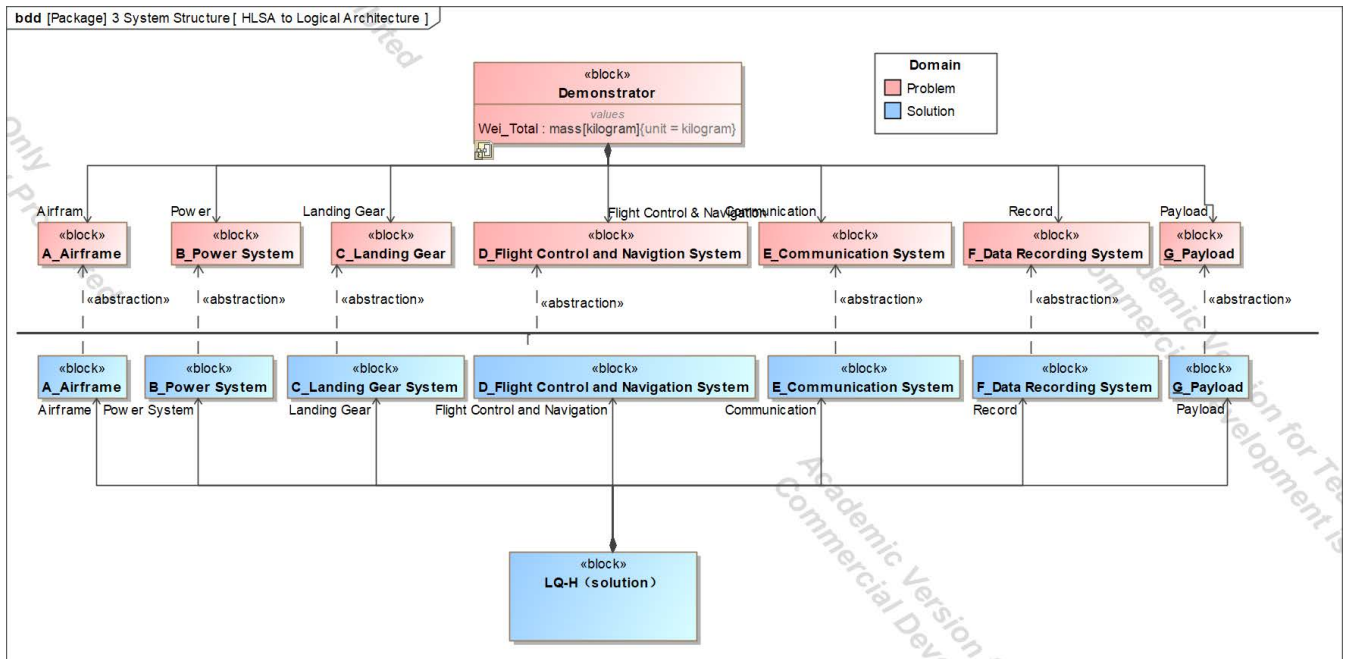


FIGURE 11. The high-level solution architecture (HLSA) model of LQ-H.

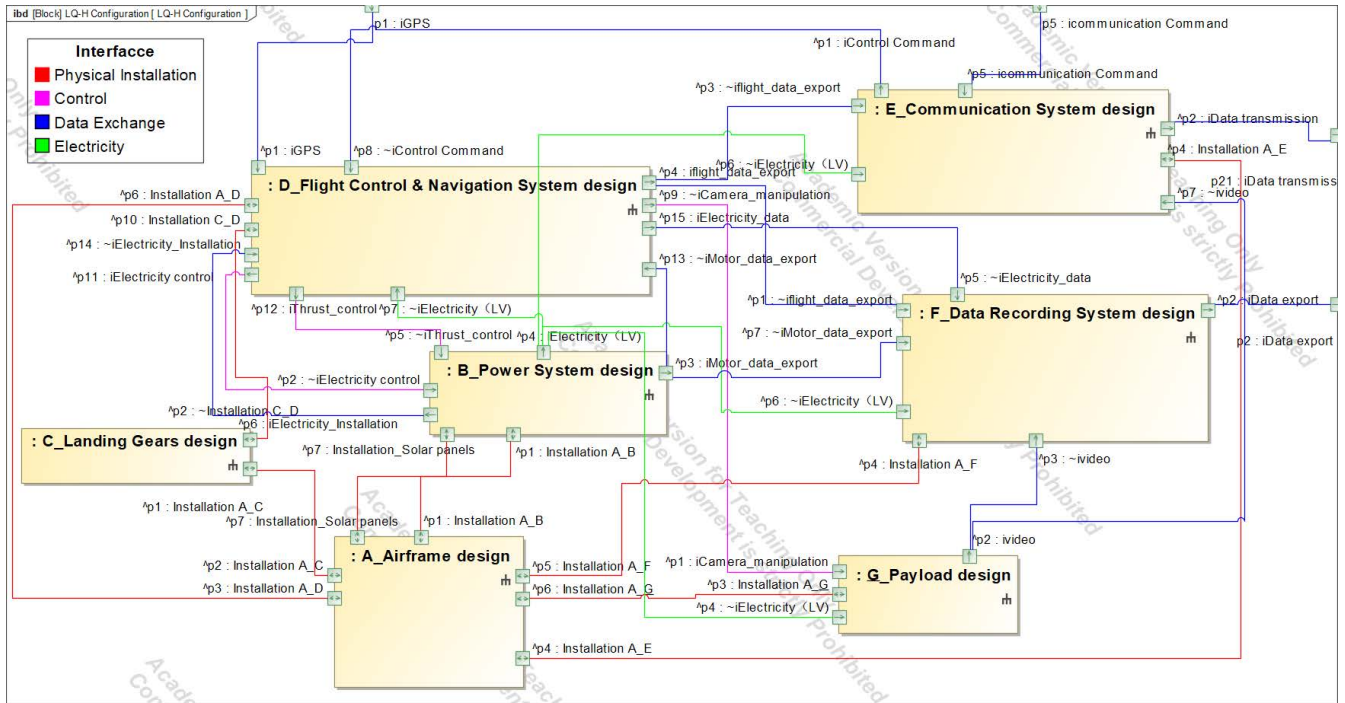


FIGURE 12. The IBD of LQ-H.

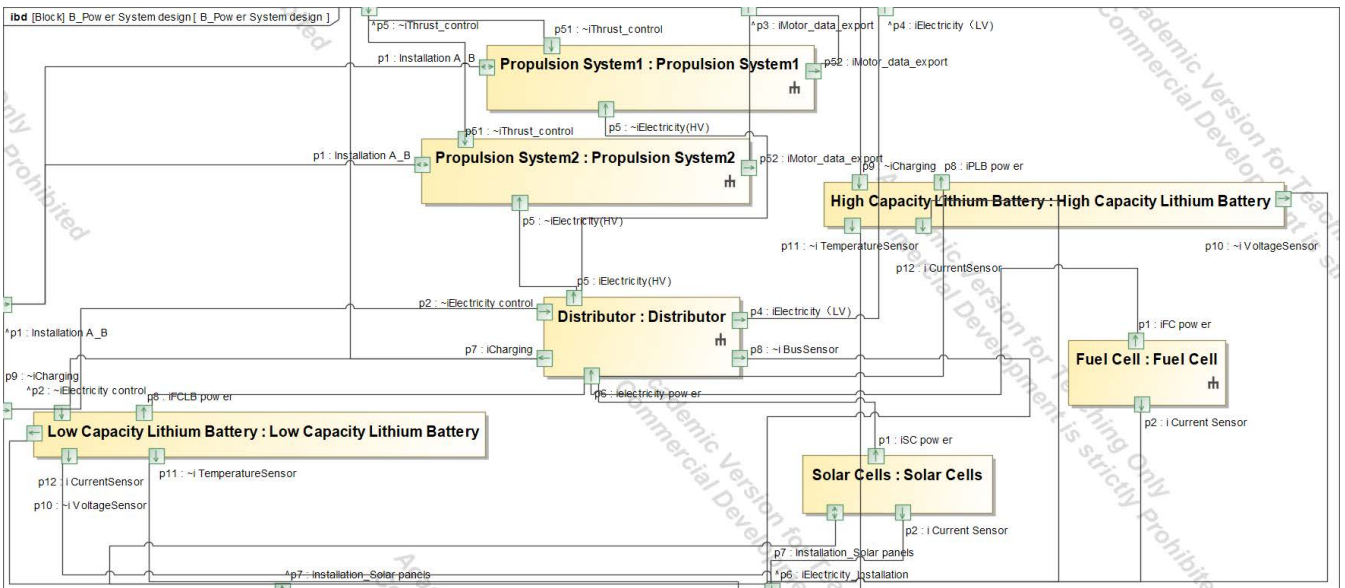


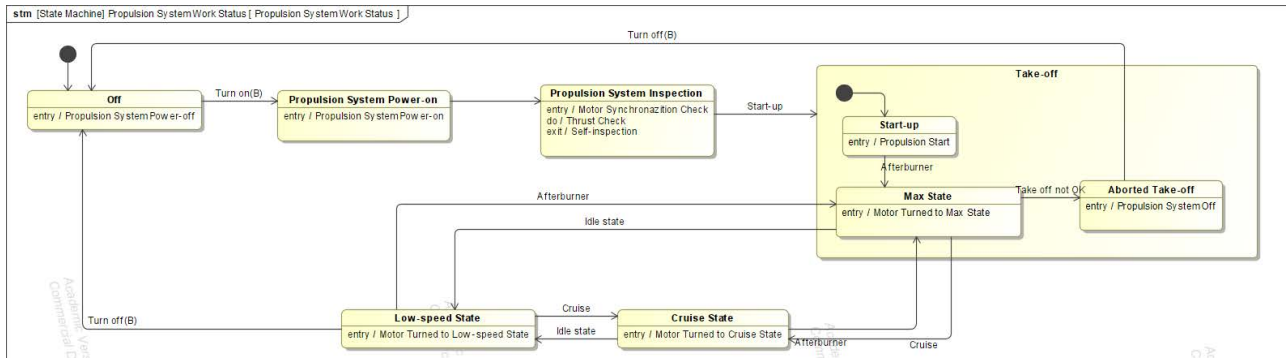
FIGURE 13. The IBD of the power system.

same requirement item. For example, in the LQ-H requirement analysis, we convert “LQH-DRO-0003.4 Thrust-to-Weight Ratio” and “LQH-DRO-0006.2 Maximum Take-off Weight” to “LQH-AFRD-0002.1 Maximum Thrust” and “LQH-NFRD-0006 Demonstrator Weight Limit”, making it easier to verify the requirement for thrust.

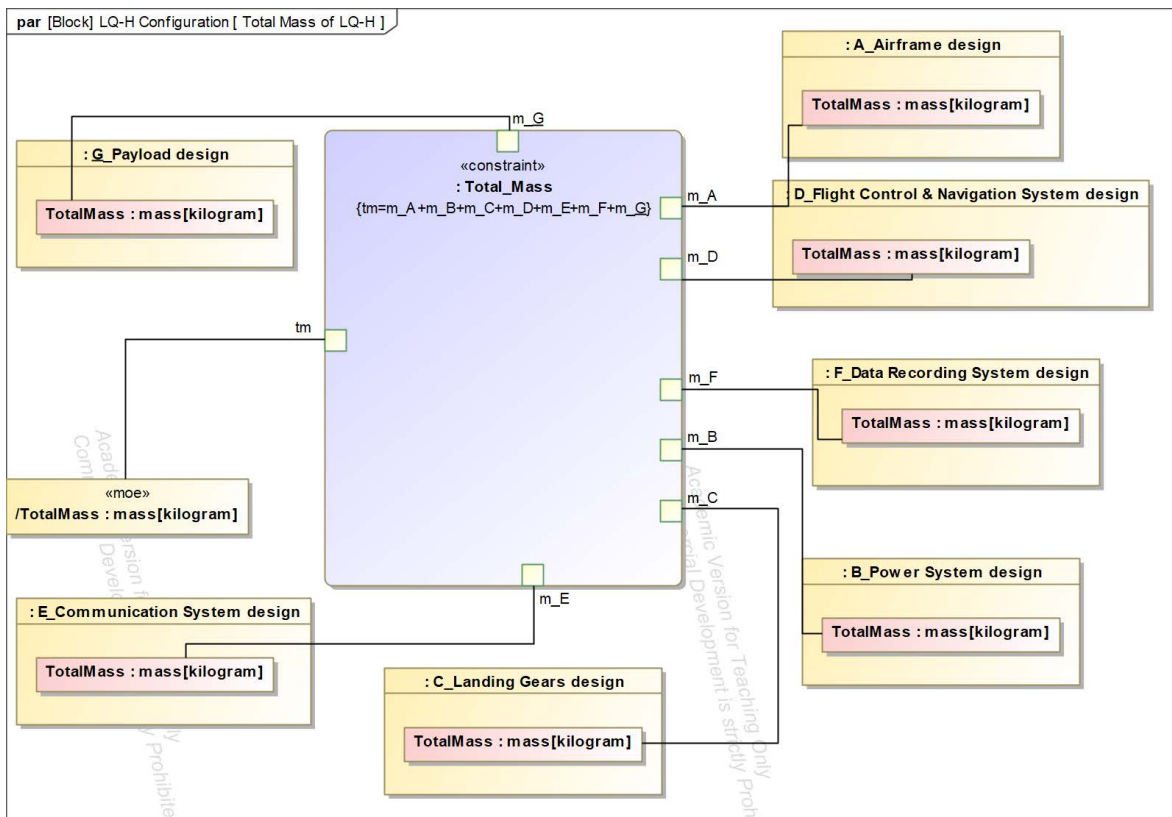
There are 167 items in LQ-H AFRD and NFRD. Table 4 shows the parent items of AFRD and NFRD.

#### D. MEASUREMENTS OF EFFECTIVENESS

Measurements of Effectiveness are indicators used to measure overall stakeholder satisfaction on the quantified demands of stakeholders for product/system performance, safety, reliability, dispatchability, maintainability, etc. [20] MoEs are captured in the same way as stakeholder needs. After the function analysis and requirement analysis of the demonstrator, the MoEs model and its relations with the



**FIGURE 14.** Use state machine diagrams (STM) to define the behavior model of the subsystem. There are 12 STMs for the 7 subsystems of LQ-H. This diagram is the state model of the propulsion system.



**FIGURE 15.** The system and subsystem parameters related to MoEs are modeled using parametric diagrams. Here a parametric constraint model of the total weight is shown.

requirement and parameter models can be established. Fig.10 shows the maximum weight as MoEs and its links with system parameters and requirements.

**VI. DESIGN SYNTHESIS**

After requirement analysis, there comes design synthesis for the demonstrator. Since the subsystems of demonstrators use commercial-off-the-shelf products or test equipment provided by stakeholders, the design synthesis process focuses on selecting the appropriate equipment for subsystems based on system requirements and modeling subsystem

structure, behavior, and parameters. Some of the design synthesis modeling results of LQ-H are shown below.

In the system structure modeling of demonstrators, a High-Level Solution Architecture (HLSA) model should first be built based on the actual situation of the selected subsystems, as shown in Fig.11. HLSA uses Block Definition Diagram to specify the subsystems and interfaces between them. And abstraction relations are used to establish a trace between the solution domain system model and the logical subsystem. The HLSA model can indicate the corresponding subsystems of solution domain to the functions and interfaces determined

Legend		01 VoC																								
DeriveReq		Co-Users Requirements LQH-VoC-0001.1 Payload Weight LQH-VoC-0001.2 Power Sources LQH-VoC-0001.3 Payload Volume LQH-VoC-0001.4 Configuration LQH-VoC-0001.5 Payload Control LQH-VoC-0001.6 Means of Take-off a LQH-VoC-0001.7 Flight Speed LQH-VoC-0001.8 Flight Duration LQH-VoC-0002 Development Team Requ LQH-VoC-0002.1 Development Procedu LQH-VoC-0002.2 Development Period LQH-VoC-0002.3 Development Reso LQH-VoC-0002.4 Durability LQH-VoC-0003 Test Users Requirement LQH-VoC-0003.1 Transport LQH-VoC-0003.2 Flight Mission LQH-VoC-0003.3 Flight Data LQH-VoC-0003.4 Pilot LQH-VoC-0003.5 Flight Safety LQH-VoC-0004 Airport & Environment LQH-VoC-0004.1 Sidewind Speed LQH-VoC-0004.2 Airport LQH-VoC-0004.3 Airspace LQH-VoC-0004.4 Runway LQH-VoC-0004.5 Wind LQH-VoC-0004.6 Headwind LQH-VoC-0004.7 Operating Temper LQH-VoC-0004.7.1 Lowest Operati LQH-VoC-0004.7.2 Highest Operati																								
02 DRO		1	4	1	1	4	1	2	1	1	1	3	1	16	10	14	21	1	1	2	3	1	1	1	1	
LQH-DRO-0001	Safety Requirements												1		1	1	4								1	1
LQH-DRO-0002	Flight Data Collection, Recording and T		2			1									6	8	7	8				1				
LQH-DRO-0003	Flight Performance Requirements		1					2	1						2		2				1		3			
LQH-DRO-0004	Power Unit Requirements			1											1		1									
LQH-DRO-0005	System Function Requirements								1							5	2	5	3					1	1	
LQH-DRO-0006	Airframe Requirements		1		1	1									1											
LQH-DRO-0007	Manufacture Requirements											1	4													
LQH-DRO-0009	Maintenance Requirements												1	2												
LQH-DRO-0010	Cost and Funding Requirements												1													
LQH-DRO-0011	User Requirements																									

FIGURE 16. Derive matrix of DRO to VoC.

Legend		02 DRO																							
DeriveReq		LQH-DRO-0001.1 Ser LQH-DRO-0001.2 LQH-DRO-0001.2 LQH-DRO-0001.2 LQH-DRO-0001.2 LQH-DRO-0001.3 LQH-DRO-0001.3 LQH-DRO-0001.4 Air LQH-DRO-0001.5 Rel LQH-DRO-0001.6 LQH-DRO-0001.6 LQH-DRO-0001.6 LQH-DRO-0002.1 LQH-DRO-0002.1 LQH-DRO-0002.1 LQH-DRO-0002.1 LQH-DRO-0002.1 LQH-DRO-0002.2 LQH-DRO-0002.2 LQH-DRO-0002.2 LQH-DRO-0002.3 LQH-DRO-0002.3 LQH-DRO-0002.4 LQH-DRO-0002.4 LQH-DRO-0002.4 LQH-DRO-0002.4 LQH-DRO-0002.5 LQH-DRO-0002.5 LQH-DRO-0002.5																							
01 AFRD		14	2	2	2	3	3	2	2	13															
LQH-AFRD-0002	Power System							1																	
LQH-AFRD-0004	Data Collecting &	10				1				13															
LQH-AFRD-0001	Flight Preparation													1	1	1	1	1	3	3	3	1	2	2	1
LQH-AFRD-0003	Installation & Fu																								
LQH-AFRD-0005	Ground Taxiing																								
LQH-AFRD-0006	Data Transmission			2	2	2	2	2	2													1	1	1	1
LQH-AFRD-0007	Flying Status & F		4				2																		
LQH-AFRD-0008	Flight Mode Trans																								
LQH-AFRD-0009	Post-flight											1						1							1
LQH-AFRD-0010	Flight Speed																								
02 NFRD		1					2	2	1																
LQE-NFRD-0001	Take-off & Landi																								
LQE-NFRD-0005	Reliability	1																							
LQE-NFRD-0003	Cost & Funding																								
LQE-NFRD-0004	Maintainability																								
LQH-NFRD-0006	Weight Limit																								

FIGURE 17. Part of the derive matrix of AFRD and NFRD to DRO.

in the problem domain. Then an IDB as shown in Fig.12 is used to model the structure of LQ-H according to the structure and interfaces of the subsystems chosen. Since the solution of LQ-H completely matches the logical subsystems, the IBD of LQ-H can be referred to Fig. Only several interfaces are adjusted based on the devices selected.

The subsystem structure modeling uses IBDs to describe the structures of the 7 subsystems. For each subsystem, there components and interfaces are modeled. The results of the power system are shown in Fig.13. The power system has two propulsion systems with propellers and motors, a distributor

controlling the supply of the electric power, and the power source consisting of two batteries, a fuel cell, and a set of solar cells.

The systems and subsystems behavior modeling use the state machine diagram (STM) to define the behavior model of the subsystems. 12 STMs are established in this research for 6 subsystems of LQ-H. Only the airframe subsystem has no STM, since the airframe of LQ-H has no functional state changes. Fig.14 shows the STM of propulsion system status. The propulsion systems of LQ-H have 6 status: off, power-on, inspection, take-off, cruise, and low-speed.

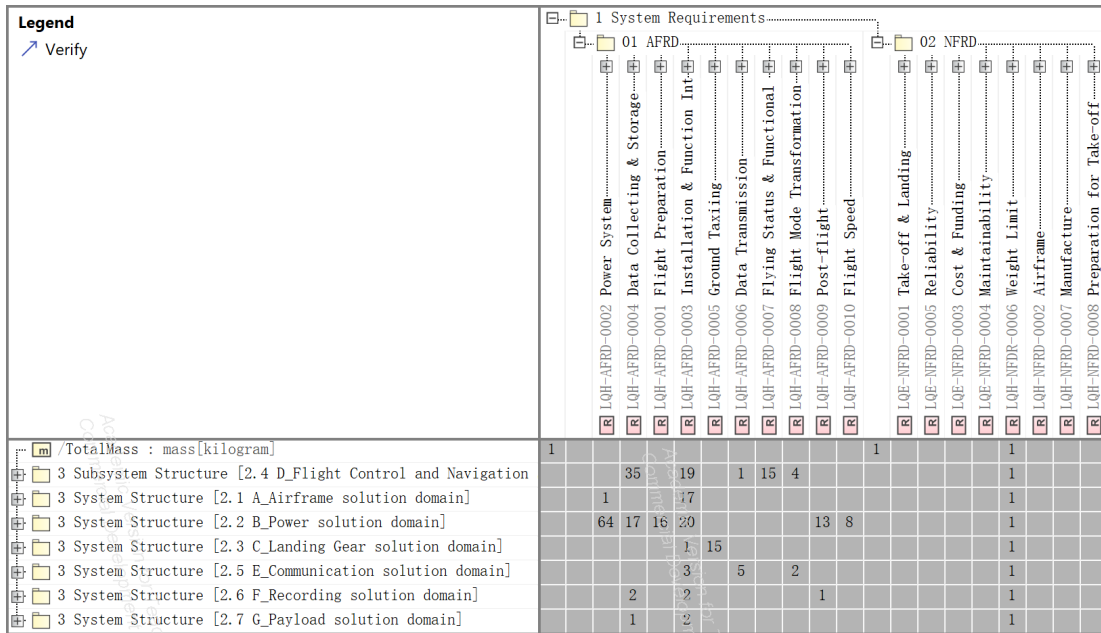


FIGURE 18. Verification matrix of functions and MoEs to system requirements.

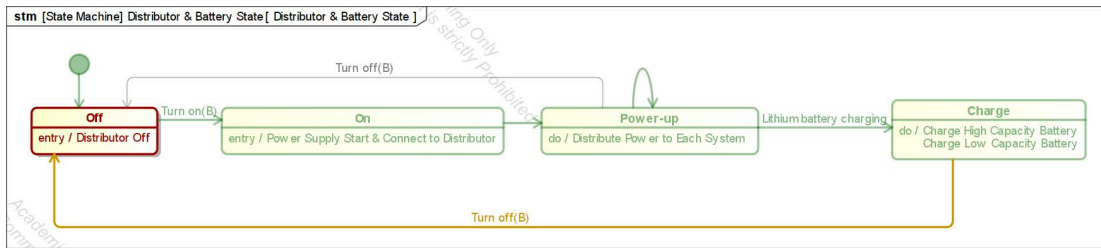


FIGURE 19. Simulation result of distributor and battery STM.

The final step of the design synthesis is the modeling of systems and subsystems parameters. Parametric Diagrams (PAR) are used to model the parameters and related MoEs. Fig.15 shows the PAR of total mass, in which the constrain of total mass connecting the total mass of each subsystems with the corresponding MoEs.

VII. MODELING RESULT ANALYSIS  
A. REQUIREMENTS TRACEABILITY CHECK

A well-developed requirement model should be traceable. Each requirement should be traced back to the corresponding source. Each requirement should be traced back to a stakeholder need layer-by-layer, and each stakeholder need should also be traced back to the corresponding stakeholder. A derive matrix is used to check the traceability of the requirements. The derive matrix of DRO to VoC is shown in Fig.16. As it is shown that each item in the VoC is derived by the corresponding DRO items, and each item of the DRO derives from a particular VoC item.

The same approach can be used to check the derivation of system requirements to DRO, as shown in Fig.17. Both

AFRD and NFRD can be traced to DRO. And every DRO item derives to the system requirements.

B. REQUIREMENTS VERIFIABILITY CHECK

Requirements verifiability check confirms that each requirement is met by the corresponding function or architecture. A verification matrix can be drawn to check the behavioral and structural design of the system corresponding to each requirement. Fig.18 shows that all requirements of AFRD can be verified by the attributes of the demonstrator structure model of the solution domain. NFRD is the non-functional requirement of the system, and in this study, a MoEs is set for the weight limit, which can be verified by the parameters of the system.

C. SYSTEM LOGIC SIMULATION

Logic simulation of the solution domain structure and behavior models can verify that there are no logic errors in the interface and state flows of the demonstrator. The logic simulation calls the IBDs and STMs of the demonstrator system and

**TABLE 5. Main parameters comparison between LQ-H and BWB UAV.**

	LQ-H	BWB UAV
Wingspan	6 meters	5 meters
Cruise Speed	28 m/s	36 m/s
Takeoff Weight (Max)	40 kg	50 kg
Development Cycle	5 months	12 months

subsystem. And the simulation result shows that there are no errors and conflicts in the design of the LQ-H demonstrator.

## VIII. CONCLUSION

In this paper, the technical process for civil aircraft scaled demonstrator is summarized according to past demonstrator development experience of BATRI. And the MagicGrid approach for demonstrator requirement analysis and architecting is proposed from the combination of the technical process and MBSE methodology. The analysis of the modeling results shows that the resulting requirement models meet the requirement modeling standard and that the architecture is logically implementable.

With the application of MBSE method to demonstrator design, the development cycle of LQ-H has been shortened to 5 months from project launch to maiden flight. As a comparison, a demonstrator as in [22] took 12 months of development cycle. The demonstrator is a blended wing body (BWB) UAV developed without MBSE approach. The BWB UAV has its wingspan, maximum take-off weight, and cruise speed similar with LQ-H, bringing similar development difficulty for these two aircrafts. Table 5 shows the characteristic parameters and development cycle of LQ-H and BWB UAV. The result shows that the MBSE approach has potential to significantly shorten the development cycle of civil aircraft demonstrators.

In conclusion, there is feasibility to apply the MagicGrid approach to the development of civil aircraft scaled demonstrators. And this approach is compatible with the previous non-MBSE demonstrator technical process, which facilitates the application of the MBSE method in the development of demonstrators. In order to accommodate the complex requirement analysis in demonstrator development, the approach in this paper divides the stakeholder needs in MagicGrid into stakeholder needs (VoC) and stakeholder requirements (DRO). The modeling practice shows that this adjustment is helpful to get verifiable and traceable demonstrator requirements. The stakeholder identification, some of the demonstrator functional scenarios, and some of the structure and behavior models in the paper are generic and can guide the future MBSE practice of demonstrator development. In addition, the requirement modeling standard summarized in this paper can be used for other complex product requirement analysis studies in the future. Meanwhile, this paper also leaves some limitations and shortcomings for further research:

1) An investigation of various MBSE methodologies (as in [10]) and modeling tools (as in [23]) will be helpful to find the best approach for civil aircraft scaled demonstrators. For future work, we will apply MBSE into various demonstrators and compare the effectiveness of different tools and methods.

2) Trade-off analysis is utilized commonly in MBSE researches. In this paper, a model-based trade-off is missing since the parameter model of subsystems is not fine enough. For future work, we will construct a model library with more detailed subsystem models and perform a model-based trade-off analysis to find optimal solution domain design for specific stakeholder needs.

## REFERENCES

- [1] *Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities*, 3rd ed., INCOSE, San Diego, CA, USA, 2011.
- [2] *Systems Engineering Vision 2020*, INCOSE, Int. Council Syst. Eng. (INCOSE), Tech. Oper. INCOSE-TP-2004-004-02, San Diego, CA, USA, Sep. 2007.
- [3] P. A. Jansma and R. M. Jones, "Advancing the practice of systems engineering at JPL," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2006, pp. 1–19, doi: [10.1109/AERO.2006.1656171](https://doi.org/10.1109/AERO.2006.1656171).
- [4] S. R. Hirshron, *NASA Systems Engineering Handbook*. Washington, DC, USA: NASA, 2017. [Online]. Available: [https://www.nasa.gov/sites/default/files/atoms/files/nasa\\_systems\\_engineering\\_handbook\\_0.pdf](https://www.nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook_0.pdf)
- [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.*, Big Sky, MT, USA, Mar. 2016, pp. 1–14, doi: [10.1109/AERO.2016.7500604](https://doi.org/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.*, Big Sky, MT, USA, Mar. 2013, pp. 1–13, doi: [10.1109/AERO.2013.6496855](https://doi.org/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, "11.5.1 Early formulation model-centric engineering on NASA's Europa mission concept study," in *Proc. INCOSE Int. Symp.*, vol. 22, no. 1, Jul. 2012, pp. 1695–1710, doi: [10.1002/j.2334-5837.2012.tb01431.x](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1109/ACCESS.2019.2952889).
- [11] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. Burlington, MA, USA: Elsevier, 2008.
- [12] J. Estefan, "Survey of model-based systems engineering (MBSE) methodologies," Int. Council Syst. Eng., San Diego, CA, USA, Tech. Rep. INCOSE-TD-2007-003-02, Jan. 2008.
- [13] A. Aleksandraviciene and A. Morkevicius, *MagicGrid Book of Knowledge*, 1st ed. Kaunas, Lithuania: No Magic, 2018.
- [14] A. Sobron, D. Lundström, and P. Krus, "A review of current research in subscale flight testing and analysis of its main practical challenges," *Aerospace*, vol. 8, no. 3, p. 74, Mar. 2021, doi: [10.3390/aerospace8030074](https://doi.org/10.3390/aerospace8030074).
- [15] J. A. Lechniak and J. E. Melton, "Manned versus unmanned risk and complexity considerations for future mid-sized X-planes," in *Proc. AIAA Flight Test. Conf.*, Denver, CO, USA, Jun. 2017, p. 3650, doi: [10.2514/6.2017-3650](https://doi.org/10.2514/6.2017-3650).
- [16] G. Galinski, J. Hajduk, M. Kalinowski, M. Wichulski, and Ł. Stefanek, "Inverted joined wing scaled demonstrator programme," in *Proc. 29th Congr. Int. Council Aeronaut. Sci.*, 2014, p. 10.
- [17] J. Bartasevicius, S. J. Koeberle, D. Teubl, C. Roessler, and M. Hornung, "Flight testing of 65 Kg T-flex subscale demonstrator," in *Proc. 32nd Congr. Int. Council Aeronaut. Sci.*, 2021, pp. 1–16. Accessed: Feb. 9, 2022. [Online]. Available: <https://mediatum.ub.tum.de/1639237>



- [18] D. Jacques, "The use of MBSE and a reference architecture in a rapid prototyping environment," presented at the MBSE Cyber Exper. Symp., Allen, TX, USA, May 2019. [Online]. Available: <https://mbsecyberexperience2019.com/speakers/abstracts/item/the-use-of-mbse-and-a-reference-architecture-in-a-rapid-prototyping-environment>
- [19] F. Torrigiani, S. Deinert, M. Fioriti, F. Di Fede, A. Jungo, L. Pisu, F. Sanchez, S. Liscouet-Hanke, P. D. Ciampa, and B. Nagel, "An MBSE certification-driven design of a UAV MALE configuration in the AGILE 4.0 design environment," in *Proc. AIAA AVIATION FORUM*, Aug. 2021, p. 3080, doi: [10.2514/6.2021-3080](https://doi.org/10.2514/6.2021-3080).
- [20] D. He, Y. Zhao, and Z. Qian, *COMAC Systems Engineering Manual*. Shanghai, China: Shanghai Jiao Tong University Press, 2016.
- [21] H. Li and S. Ying, "The top level aircraft requirements capture experiences summary," (in Chinese), *Civil Aircr. Design Res.*, vol. 120, no. 1, pp. 89–92, 2016, doi: [10.19416/j.cnki.1674-9804.2016.01.025](https://doi.org/10.19416/j.cnki.1674-9804.2016.01.025).
- [22] D. Thompson, J. Feys, M. Filewich, S. Abdel-Magid, D. Dalli, and F. Goto, "The design and construction of a blended wing body UAV," in *Proc. 49th AIAA Aerosp. Sci. Meeting Including Horizons Forum Aerosp. Expo.*, Orlando, FL, USA, Jan. 2011, p. 841, doi: [10.2514/6.2011-841](https://doi.org/10.2514/6.2011-841).
- [23] M. Rashid, M. W. Anwar, and A. M. Khan, "Toward the tools selection in model based system engineering for embedded systems—A systematic literature review," *J. Syst. Softw.*, vol. 106, pp. 150–163, Aug. 2015, doi: [10.1016/j.jss.2015.04.089](https://doi.org/10.1016/j.jss.2015.04.089).



**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.



**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 research interests include aircraft conceptual design, digital twin, model-based systems engineering, and big data.



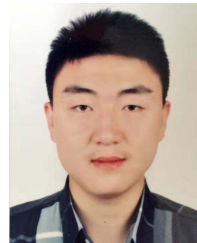
**CHI ZHANG** received the B.Eng. degree in electromechanical engineering from the University of Southampton, U.K., and the M.Sc. degree in aerospace vehicle design from Cranfield University, U.K. He is currently the Project Lead Engineer at the COMAC Beijing Aircraft Technology Research Institute (BATRI).



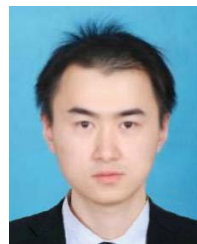
**YONGZHI HUA** was born in Henan, China, in 1988. He received the B.E. degree in mechanical engineering and automation from Beihang University, Beijing, China, in 2010, and the M.E. degree in mechanical manufacturing and automation from Tsinghua University, Beijing, in 2013. He currently works at China Academy of Launch Vehicle Technology. His main research interests include aircraft overall design, structural optimization design, and tolerance design.



**RAN XIE** was born in Anhui, China, in 1996. She received the B.E. degree in aircraft design from the Nanjing University of Aeronautics and Astronautics, in 2018, and the M.E. degree in hydrodynamics from the School of Aeronautic Science and Engineering, Beihang University. She currently works at the Yangzhou Collaborative Innovation Research Academy, Shenyang Aircraft Design and Research Institute. Her main research interests include aircraft conceptual design, multi-scheme decision making, and model-based systems engineering.



**RUNXI YU** was born in Shaanxi, China, in 1993. He received the B.E. degree in aircraft design and engineering from Beihang University, Beijing, China, in 2016, and the M.E. degree in aerospace engineering from the Georgia Institute of Technology, Atlanta, CA, USA, in 2018. He currently works at the COMAC Beijing Aircraft Technology Research Institute. His main research interests include flight dynamics, feedback control of dynamic systems, aircraft conceptual design, and model-based systems engineering.



**CHEN HUANG** was born in Hubei, China, in 1989. He received the B.E. and Ph.D. degrees in aircraft design from Beihang University, in 2011 and 2019, respectively. He currently works at the COMAC Beijing Aircraft Technology Research Institute. His main research interests include aircraft conceptual design, aircraft systems, and model-based systems engineering.

...