applied sciences

MDPI

Article

Design and Application of Logical Range Framework Based on Digital Twin

Hao Chen [1,2], Zheng Dang [1,2], Xinhong Hei [1,2,*] and Kai Wang [3]

1 School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China; chenhao@xaut.edu.cn (H.C.); zheng.dang@outlook.com (Z.D.)
2 Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China
3 Xi'an Institute of Measurement and Testing Technology, Xi'an 710068, China; wangkai_625@163.com
* Correspondence: heixinhong@xaut.edu.cn

Abstract: With the development of the concept of integrated flight testing, joint testing has become a key research trend. The use of a Logical Range can allow one to overcome the shortcomings of traditional test ranges, allowing for full use of the resources of each range when implementing joint flight testing. However, the Logical Range concept also has problems, such as those relating to the low reusability of test resources and insufficient ability to monitor the resource status in real-time. Considering such problems, a Logical Range framework based on digital twin technology is proposed in this paper. On the basis of this framework, a Logical Range system based on a digital twin for the real-time mapping of real physical behavior is constructed and the three key technologies used in the system are detailed. The feasibility of the framework is successfully verified through a flight test, demonstrating that the proposed framework can provide key support for the application of digital twin technology in the field of flight testing.

Keywords: logical range; flight test; joint test; digital twin

check for updates

Citation: Chen, H.; Dang, Z.; Hei, X.; Wang, K. Design and Application of Logical Range Framework Based on Digital Twin. *Appl. Sci.* **2023**, *13*, 6589. https://doi.org/10.3390/app13116589

Academic Editors: Zhihan Lv, Kai Xu and Zhigeng Pan

Received: 22 March 2023
Revised: 25 May 2023
Accepted: 26 May 2023
Published: 29 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

From the conceptual development of integrated flight testing emerges the concept of joint testing. A joint test can realize the seamless connection of cross-regional proving grounds, complete the interconnection between different test systems, and realize the data sharing, reusability, and combinability of test resources [1,2]. In order to implement joint testing across a range, it is necessary to connect multiple cross-area range resources and facilities, from which the concept of Logical Range arises. A Logical Range refers to the aggregation of test resources without geographical boundaries, across various ranges and facilities [3]. Use of the Logical Range concept can overcome the shortcomings of traditional test ranges, such as geographical division, single capability, and difficulties in completing certain comprehensive test tasks. It allows for the connection of many different ranges and test resources in a network, such that these ranges distributed in different geographical locations logically form a comprehensive range, allowing for full use of the facilities and resources of each range [4].

However, there are two main problems in existing Logical Range test processes: First, the reusability of test resources is low. For test resources that have a simple functional design but short life or those which are single-use only, a Logical Range cannot realize their long-term reuse. Second, the real-time resource status monitoring ability is insufficient. There are many types and quantities of test resources in the joint testing process. As the goal of each test is different, the test resources included also differ, which leads to existing Logical Range approaches being unable to monitor the status of all test resources in real-time, thus rendering the joint test inefficient.

With the development of simulation and computer technology, Digital Twin technology has played increasingly important roles in the fields of digital workshops [5–7], digital cities [8–10], and intelligent manufacturing [11–15]. Digital twin technology provides a link between physical space and virtual space. Moreover, it can make full use of physical models, real-time data, and historical operation and maintenance data, allowing for the combination of multi-variable, multi-scale, and multi-probability computer simulation processes to construct twins of physical entities in virtual space. Further, it can also reproduce all of the states of physical entities in real-time, providing high fidelity and high integration and, thus, reflecting the whole life-cycle process of the entity and its state [16].

The emergence of Digital Twin technology provides an effective solution to the above-mentioned problems related to the Logical Range. One of the goals of Logical Range systems based on the use of a digital twin is to realize the real-time digital mirroring of range resources and flight test processes, as well as the data-driven real-time state synchronization mapping of virtual and real ranges through the construction of a multi-dimensional high-fidelity range dynamic model. Another aim of these systems is to combine artificial intelligence algorithms for data analysis, mining, fault diagnosis, and intelligent decision making, thus realizing the transparency, efficiency, and intelligence of the range. The digital twin of the test resource is constructed in the virtual scene, and its operation logic is learned through the artificial intelligence algorithm during the test process. In some scenarios, the twin is used to replace the real test resources, in order to improve the reuse rate of resources. The insufficient real-time test resource status-monitoring ability in the Logical Range can be resolved by observing the state change of the twin of the physical test resource in real-time during the test process.

In recent years, scholars at home and abroad have conducted in-depth research on the theoretical concept and model construction of digital twins in the fields of aerospace and flight testing. In terms of research on digital twin technology in the aerospace field, some scholars [17–19] have conducted in-depth research on digital twin modeling methods for aerospace parts and components, while other scholars [20–22] have studied the specific application of digital twins in the aerospace field, as well as designing methods and frameworks to realize and apply digital twin technology. In view of the existing research on digital twins in the field of flight testing, Liu Yu et al. [23] proposed the application prospect of six digital twins for flight testing, including the development and overall planning of flight test requirements, flight test task design, integrated testing and modification, flight test organization and implementation, rapid maintenance support, and flight test engineer training. Liu Yan et al. [24] designed a digital twin flight test simulation platform which can simulate flights in a virtual scene. Compared with the real flight test results, the virtual flight test on the platform can meet the real-time and stability requirements of the flight. In the research of digital twin framework, Guo Jutao et al. [25] designed an aerospace digital twin workshop framework and elaborated on the application of digital twin in aerospace intelligent manufacturing scenarios. This framework provides a feasible technical approach for workshop production control in military enterprises. Chen Jiang et al. [26] proposed a three-dimensional real-time monitoring system framework for workshop production status for digital twin, and detailed several key technologies in the framework implementation process. Finally, the feasibility and effectiveness of the proposed framework were verified by a seal production workshop.

Although Liu Yu et al. [23] have studied the application of digital twin in the field of flight testing, their research only proposes some prospects for the application of digital twin technology in flight testing, without elaborating and solving problems encountered during the implementation process. Although the flight test simulation platform designed by Liu Yan et al. [24] can meet the experimental requirements of virtual flight tests, its platform cannot meet the requirements of cross-regional joint flight tests. The digital twin framework proposed by Guo Jutao et al. [25] and Chen Jiang et al. [26] is applicable to aerospace production workshops and seal production workshops, respectively. Due to the lack of connectivity between the flight test process and the production process in the production workshop, the equipment used in the flight test and production workshop are also different. The framework proposed by the above two scholars cannot be applied to cross-regional flight testing. In addition, the proposed framework cannot meet the need for dynamic expansion of equipment during cross-regional flight testing.

In summary, scholars have made significant achievements regarding the use of Digital Twin technology to solve problems in the corresponding fields. However, there have been few studies on the construction of digital twin frameworks, key technologies, and test process monitoring for flight testing using the Logical Range concept. Therefore, in order to use Digital Twin technology to solve the problems associated with existing Logical Range approaches, as well as to satisfy the actual needs of Logical Range construction in the flight test process, we propose and implement a Logical Range framework based on digital twin technology. Furthermore, we verify the proposed framework in the context of the flight test process for an aircraft. Our key contributions are as follows:

- We analyze the problems related to existing Logical Range approaches and propose a Logical Range framework based on digital twin technology.
- We implement the proposed framework and introduce several key technologies used in the system based on the framework.
- We verify the feasibility of our proposed framework through a flight test.

The remainder of this paper is organized as follows: Section 2 introduces our proposed digital-twin-based Logical Range framework. Section 3 introduces three key technologies used in the system implemented based on the proposed framework. Section 4 introduces the implementation of the Logical Range based on digital twin technology, and details the verification of our proposed framework through an example.

2. Logical Range Framework Based on Digital Twin

According to the flight test task process and the requirements of the Logical Range virtual test, the idea of hierarchical development is adopted to construct the Logical Range framework based on digital twin technology, which is divided into a physical layer, data layer, virtual layer, service layer, and implementation layer, as depicted in Figure 1. In Figure 1, it can be seen that in order to meet the dynamic expansion of equipment and data communication requirements during cross-regional flight tests, a proxy has been introduced into our framework, which can connect hardware equipment and software systems to meet the dynamic expansion and data communication needs of the equipment. This part of the content will be detailed in subsequent chapters.

1. *Physical Layer*
   As the basis of the whole framework, the physical layer is divided into an equipment layer (E_l), range layer (R_l), and test layer (T_l), according to the application scenario and functional structure. The relationship between these layers is as follows:

$$E_l \subset R_l \subset T_l. \tag{1}$$

   The equipment layer can be regarded as the basis of the physical layer, which is composed of theodolite, high-speed camera, radar, and GPS (Global Positioning System) equipment, along with the other equipment and test resources distributed in each range. The range layer consists of ranges distributed in different regions. As a complex of test resources, the range is also the basic unit of the flight test process. The test layer consists of different test schemes. The test scheme includes all the supporting resources and information needed for a test, usually consisting of one or more ranges.
2. *Data Layer*
   Data play a crucial role in the whole system. Data flow between different parts of the system drives the operation of the whole system. The data in the data layer are mainly composed of physical layer data, virtual layer data, and service layer data. The physical layer data are mainly composed of static data and dynamic data. Static data include equipment information, geometric parameters, and so on. Dynamic data are the data collected and generated by the equipment during operation. The

data of the virtual layer are mainly composed of the data model, geometric model, behavioral model, and data generated by the virtual twin scene. The real-time data and historical data collected and generated during the operation of each model and scene in the virtual layer are part of the virtual layer data. The data of the service layer are composed of the data generated by each functional module in the service layer during operation, including the source data obtained from different databases and the data generated by various calculations in the service layer.

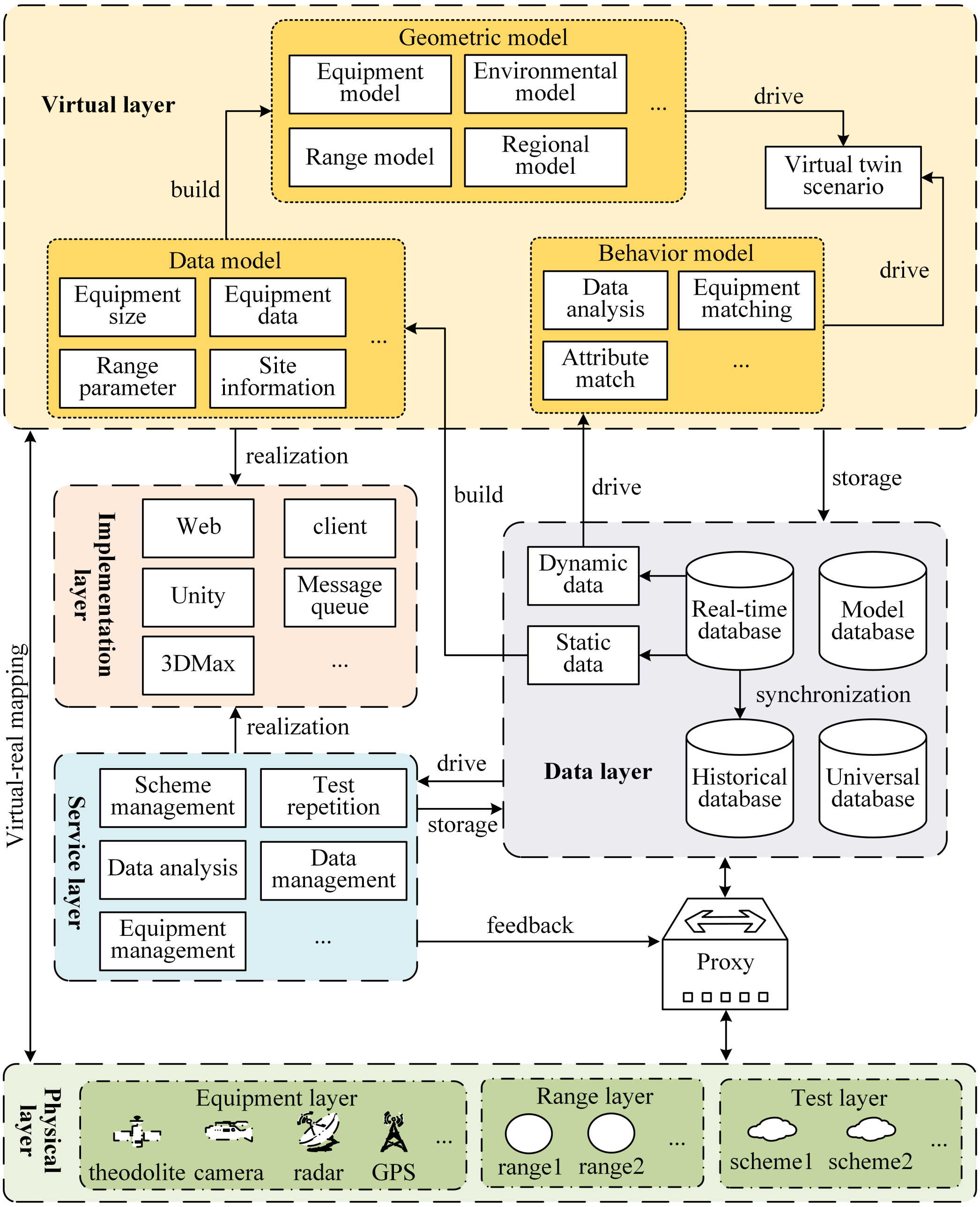

Figure 1. Logical Range Framework based on digital twin.

3. *Virtual Layer*

As the "twin" of the physical layer, the virtual layer is an indispensable part of the system. The virtual layer is mainly composed of a data model (D_m), geometric model (G_m), behavior model (B_m), and virtual twin scene (V_s). The relationship between them is given as follows:

$$((D_m \subset G_m), B_m) \subset V_s. \tag{2}$$

The data model is constructed from static data in the data layer, providing data support for the geometric model. The geometric model, supported by the data model, describes geometric information such as the appearance, structure, and size of various types of equipment and ranges. The behavior model is driven by dynamic data, through the script file, in order to achieve real-time data matching. The virtual twin scene is a virtual scene constructed under the joint action of the three models mentioned above, which can logically restore the Logical Range with high fidelity.

4. *Service Layer*
The service layer implements its corresponding functions based on the drivers of the data layer. With the support of the data layer, the service layer can realize the functions that the physical layer and the virtual layer cannot, such as all kinds of data management, visualization and analysis, test reproduction, equipment management, and so on. At the same time, the correlation calculation results of the service layer allow for feedback with the physical layer to optimize the functioning of the physical layer.
5. *Implementation Layer*
The implementation layer is required to implement the whole system using related technologies. The modeling software is used to model the equipment and the modeling results are imported into Unity 3D, in order to realize the construction and display of the equipment "twin". The data-related functions are implemented in the form of webpages through a well-designed human–machine interface and interaction mode. Both desktop clients and mobile phones can access and operate the webpages.

3. Key Technologies

3.1. Proxy-Based Equipment Dynamic Extension Method

The current Logical Range systems require the code of the system to be modified when expanding to new equipment. Moreover, the properties and functions of the new equipment cannot be predicted, and the code for them cannot be written during system development. If the system code is modified in order to add new equipment, the workload will be increased and, furthermore, the "Open Close Principle" of software design will be violated. Therefore, in order to improve the testing efficiency and realize the efficient use of test resources, it is necessary to realize dynamic expansion of the equipment. The proposed system introduces a proxy to solve this problem. The location of the proxy in the system is shown in Figure 2, between the equipment and the upper layer software. The equipment only needs to exchange data and receive instructions with the corresponding equipment proxy, without considering the various requirements of the upper software for the equipment. The proxy reduces the interference of upper software with the equipment, as well as reducing the other functions that the equipment needs to implement in order to be compatible with the upper system. As for the upper software, it only needs to communicate and interact with the equipment proxy, without considering the different communication protocols, data formats, and sampling frequencies of the various equipment.

The data content, data format, and sampling frequency of different equipment may differ, even among different brands of the same equipment. The proxy should resolve these problems, thus achieving the goal of compatibility with different equipment, improving system compatibility, and enhancing test efficiency. Therefore, a base class *Root* is designed, in order to specify the basic attributes and functions that all equipment should have. Its formal description is as follows:

$$Root = \{equipmentID, equipmentType, dataAcquisition, dataTransmission, instructionParsing\}. \quad (3)$$

It can be seen, from Equation (3), that every piece of equipment that accesses the system must have at least two *equipmentID* and *equipmentType* properties: *equipmentID* is used to bind virtual and real equipment, while *equipmentType* is used to mark the equipment type. In addition, it also needs to possess data acquisition, data transmission, and instruction parsing functions, where data acquisition means that the proxy needs to have the ability to obtain data from the equipment, data transmission means that the proxy needs to be able to send data to the upper application, and instruction parsing means that the proxy needs to parse the instructions from the upper application.

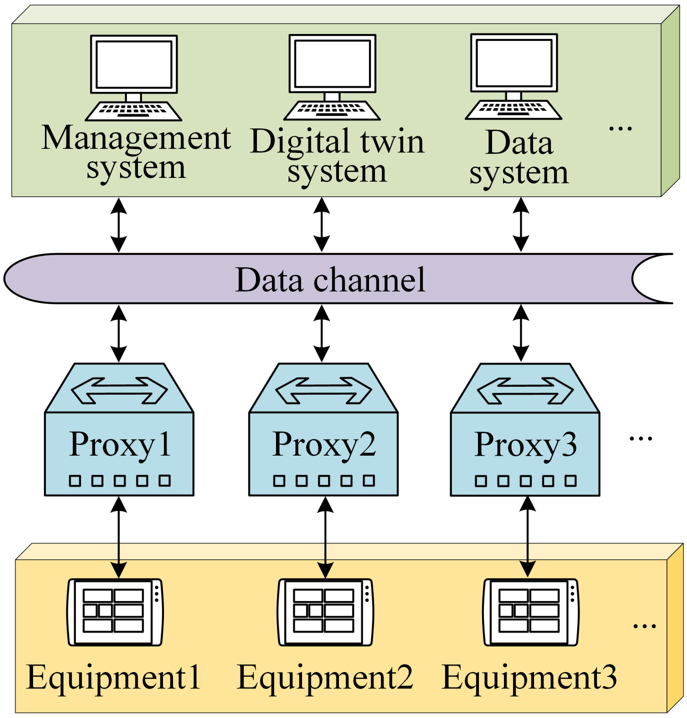

Figure 2. Location of the equipment proxy in the proposed system.

The *Root* specifies only the properties and functions that the proxy of the newly extended equipment must have. However, in fact, the properties and functions of the equipment far exceed those specified in *Root*. Therefore, the proxy needs to be extended, according to different equipment, on the basis of inheriting *Root*.

The complete design and deployment process of the proxy is shown in Figure 3. Figure 3 illustrates the proxy development and usage process using IMU (Inertial Measurement Unit) and TSPI (Time Space Positioning Information) as examples. After inheriting *Root*, the proxy only has the properties and methods declared in *Root*, which cannot facilitate the compatibility of different equipment. It also needs to be expanded according to the functions of the equipment. After the proxy is expanded, it also needs to be filled with code, in order to further improve the proxy according to the specific functions of the equipment, such that it reflects all the functions of the equipment. The proxy that completes the code fill is generic for the same equipment. The proxy is then uploaded to the proxy management platform, which conducts unified management and configuration of all proxies through an interactive interface. In the proxy management platform, a specified proxy can be selected and a proxy object can be created for it, where each proxy object matches a specific piece of equipment. Proxy objects and equipment are bound to each other through *equipmentID*. After binding, the equipment is taken over by the binding proxy object. The equipment only needs to interact with the proxy object, while the upper layer application software only needs to interact with the proxy object.

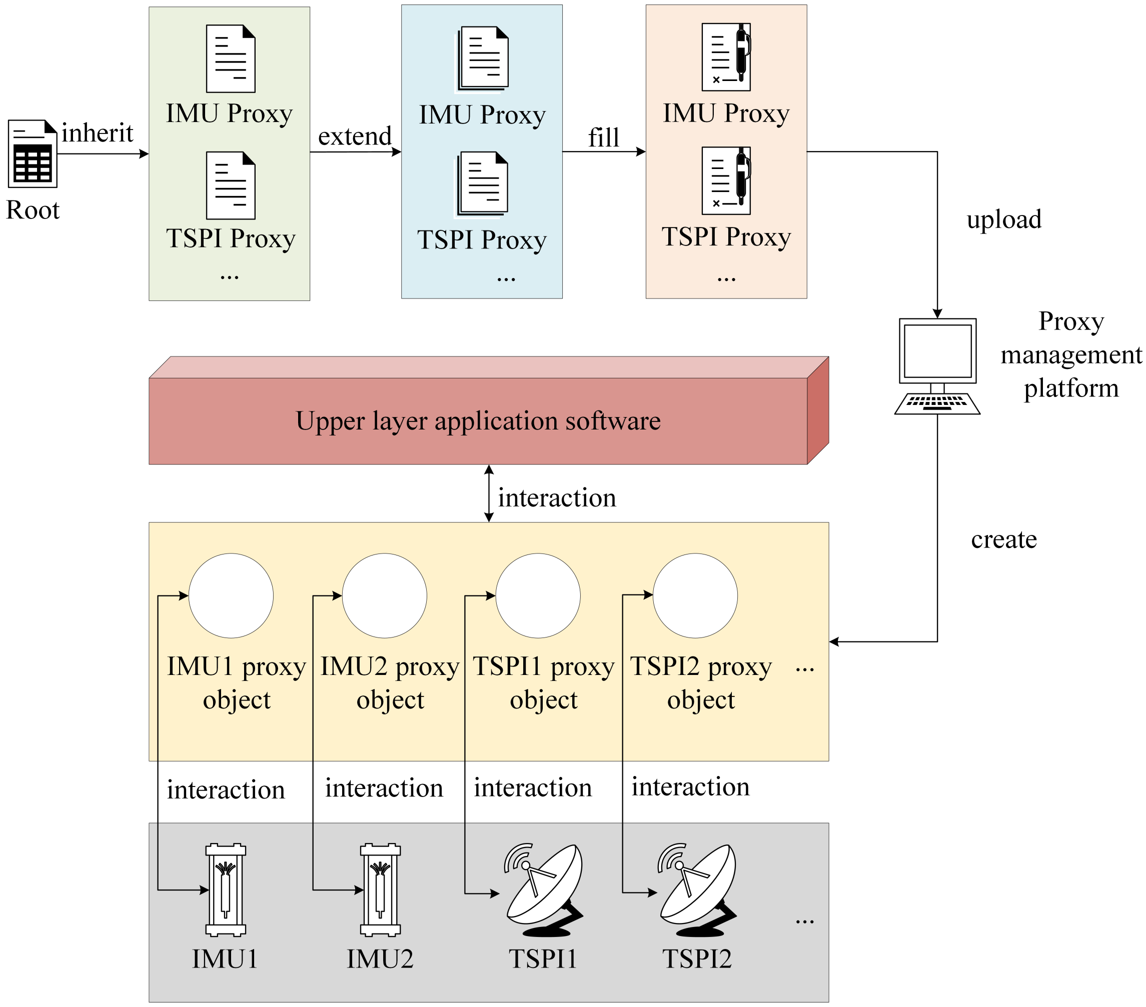

Figure 3. Schematic diagram of proxy creation and operation.

When the system needs to extend to new equipment, it needs to check whether the corresponding proxy exists for the new equipment. If the new equipment has a corresponding proxy, the proxy object is created for the equipment in the proxy management platform and the proxy object is bound to the equipment to complete the equipment expansion. If the corresponding proxy does not exist for the new equipment, the corresponding proxy needs to be developed (according to the proxy development process described above), following which the proxy is uploaded to the proxy management platform. A proxy object is then created for the new equipment, binding between the proxy object and the equipment is completed and, finally, the equipment extension is implemented. A flowchart of the process for adding new equipment is shown in Figure 4.

From the above description of the whole proxy process, it can be seen that, after adopting the proxy mechanism in this system, the intrusion to the system in the process of dynamically adjusting the physical layer equipment is minimal, and the whole process is in accordance with the "Open Closed Principle." The equipment that is added to the system will not only be limited to physical equipment, as simulation equipment can also be connected to the system through proxies to achieve system integration.

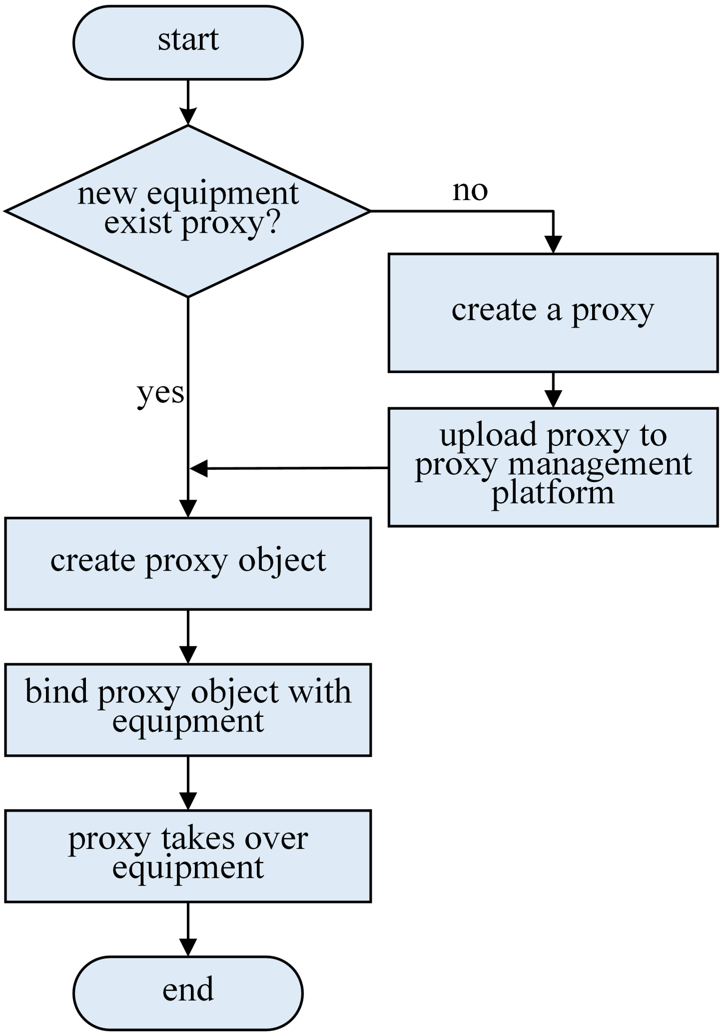

Figure 4. Equipment addition process flowchart.

3.2. Data Communication Architecture

In the flight test, multiple ranges and various equipment are required to carry out coordinated tests. Therefore, a large amount of real-time data will be generated during testing. The system needs to receive, process, calculate, and display these large amounts of real-time data, which poses huge challenges regarding the data-processing ability and network performance of the system. In order to reduce the processing pressure and network burden of the system, the data are divided into static data and dynamic data, as shown in Figure 5. Dynamic data must be generated, received, and acquired in real-time during system operations, whereas static data only need to be acquired once (e.g., during system startup). This can reduce the system load to a certain extent.

Dynamic data are the real-time data collected by each piece of equipment and sensor during the operation of the Logical Range system based on digital twin technology. Dynamic data mainly consist of two parts: the data collected by sensors (e.g., current humidity, temperature, wind speed, wind direction, and other environmental information), and the data collected and generated by each piece of equipment (e.g., the latitude and longitude, magnetic declination angle, signal-to-noise ratio, radial velocity, height, acceleration, angular acceleration, position coordinates, horizontal angle, vertical angle, semi-measured back-angle, and so on).

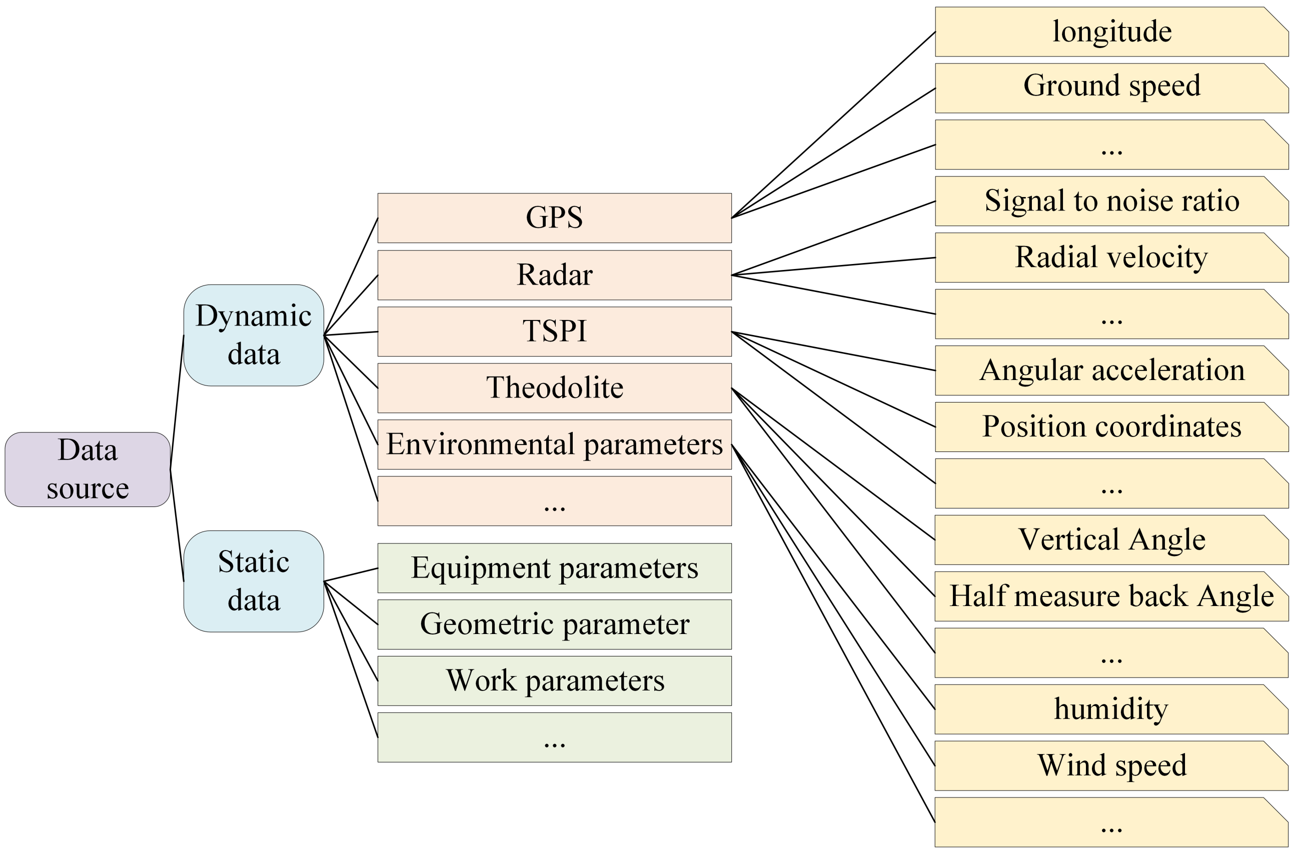

Figure 5. Data classification.

The static data are mainly the information of the physical equipment in the range, such as the basic parameters of the equipment, the geometric parameters of the equipment, the basic working parameters of the equipment, and so on. Static data are generally the basic information required for the normal operation of the equipment, which can typically be obtained from the equipment manual or the equipment use staff.

In order to realize the data collection described above, we designed the Logical Range system based on digital twin communication architecture, as shown in Figure 6. Each piece of equipment is connected to the associated equipment proxy running on the range side through a short-distance transmission protocol. The equipment proxy obtains the data from the equipment, performs simple checksum processing on the data, and stores the processed data in the local database of the range. The equipment proxy stores the data in the local database and sends the data to the cloud at the same time, storing the data in the cloud real-time database. The data are then sent to the message middleware, in order to drive the functions related to the virtual and service layers.

The Logical Range system based on digital twin technology, as the data reading client, reads the real-time data in a subscribe/publish manner to realize the real-time synchronous display of virtual and real data. At the same time, the data stored in the cloud also provide data support for data analysis, data calculation, and other functions of the service layer.

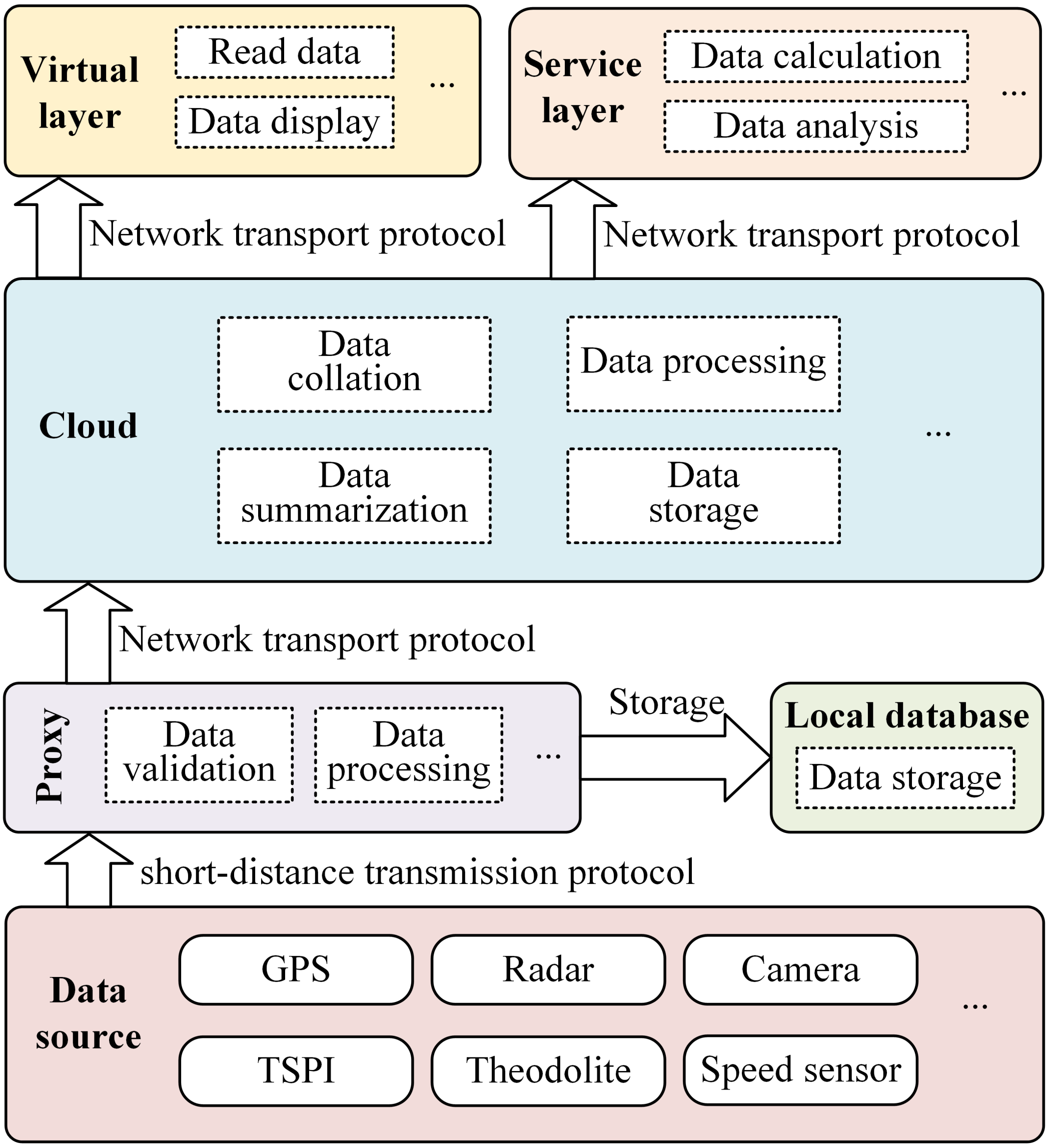

Figure 6. Data communication architecture.

3.3. Virtual–Real Synchronous Mapping Method

The mapping between virtual space and physical space is the basis for the realization of virtual–real interactions. In order to ensure the accurate characterization of the test process and to realize the effective collaboration between test resources, it is necessary to realize the virtual and real synchronization mapping between the physical and virtual test resources.

The virtual–real mapping process is mainly divided into four parts—data acquisition, data integration, data matching, and data visualization—as shown in Figure 7. The main steps are detailed in the following:

1. *Data Acquisition*
   The first step of data acquisition is to accurately collect the original data from the physical test field in real-time. The collected data will be used as the data source for the data processing and mapping phases. In order to be compatible with a variety of data transfer protocols, proxies are developed for all kinds of equipment. The proxy collects raw data through serial ports, Bluetooth, local area networks, and so on.
2. *Data Integration*
   With the real-time massive data collected during the data acquisition stage, data integration mainly involves pre-processing operations (e.g., data cleaning and con-

version), followed by the unified storage and management of data, ensuring that the value of the data is maximized.

3. *Data Matching*
In order to realize virtual–real synchronous data mapping, a subscribe/publish data transmission mode was designed based on search tree. For each piece of physical equipment, its proxy will create a topic in the message middleware during the initialization process for the transmission of equipment data, and the topic name is added to the search tree. Models in the virtual space will search the search tree for topic names before receiving data. As the topic names satisfy certain generation rules, at least one result will be retrieved and topic matching can be completed.
4. *Data Visualization*
Once the data matching is completed, the twin model in the virtual space receives real-time data from the physical space. These real-time data will drive the virtual equipment to update its state, according to the time sequence, thus completing the state update of the whole twin scene.

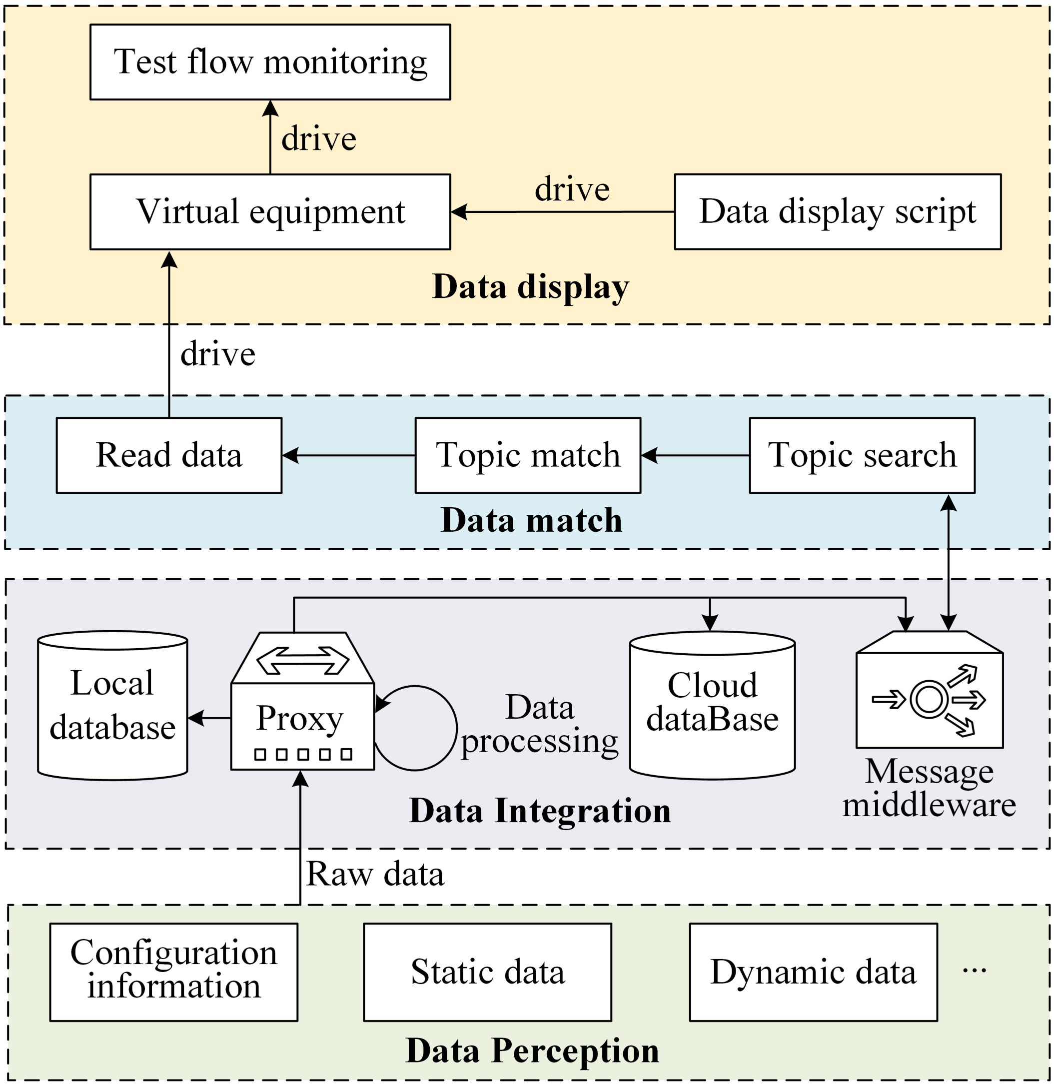

Figure 7. Virtual–real synchronization mapping process.

4. Application Examples and Analysis

Based on the above research, we implemented a system based on the proposed framework, where the system includes the functions of planning scheme construction and management, Logical Range construction and management, data integration management, unified equipment management, and so on. The system was developed through a combination of Browser/Server (B/S) and Client/Server (C/S) architectures.

According to the situation of Logical Range flight testing, the proposed system uses a combination of short-range transmission protocols and long-distance network transmission protocols to complete data transmission. Short-distance transmission protocols include Bluetooth, ZigBee, Wi-Fi, and RS-485 serial ports, among others, while long-distance transmission protocols include TCP/IP, UDP, and so on. In terms of data storage, the combination of a relational database PostgreSQL and a non-relational database MongoDB was used to realize the storage, processing, and analysis of massive structured, semi-structured, and unstructured data. In terms of data management, we used Vue.js and SpringBoot to develop a web management platform, and employed container technology to deploy the developed system on the cloud platform. The web management platform realizes data and equipment management for the entire system, including data visualization, data fusion, log management, user management, unified equipment configuration, equipment matching, and so on. The user management and system history information interface in the web management platform is shown in Figure 8.

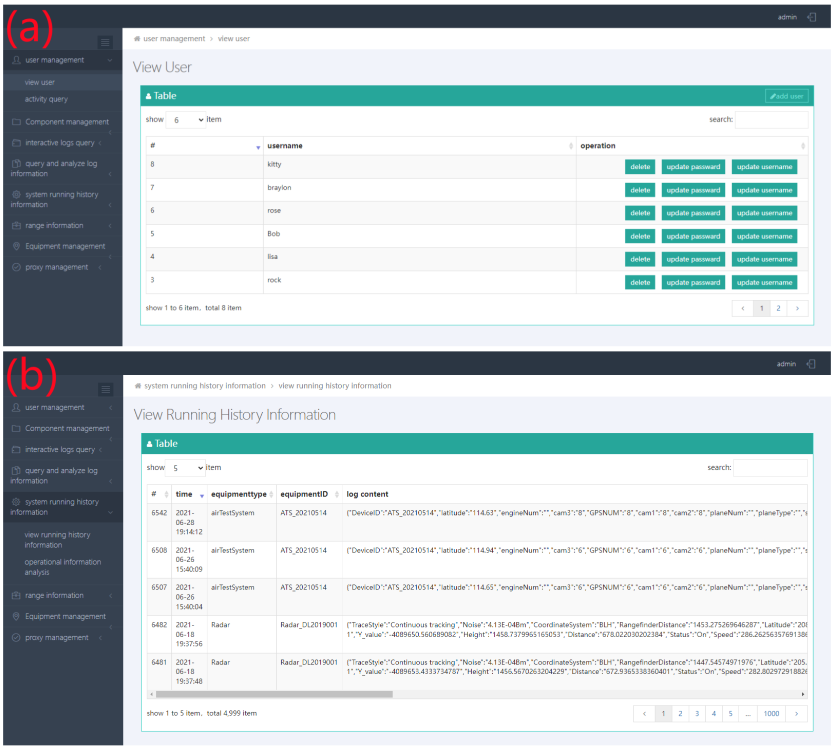

Figure 8. Web management platform interface. (**a**) User interface view (**b**) Historical information interface view.

In terms of virtual scene construction, 3DMax was used to build virtual models in various equipment and scenes, and we imported the models into Unity3D. We built the virtual twin scene in Unity3D, and used the C# script language to drive the operation of various models and components in the virtual twin scene. A C/S architecture was adopted to complete the twin system construction. The client can be installed on a user's PC, and can realize the real-time display of equipment data, planning scheme construction, and other functions. Furthermore, the platform was deployed in a cloud server using container technology to realize the functions of data interaction, analysis, and processing. The test plan management interface of the digital twin logic range platform we have built is shown in Figure 9. The overall data flow of the system is depicted in Figure 10.

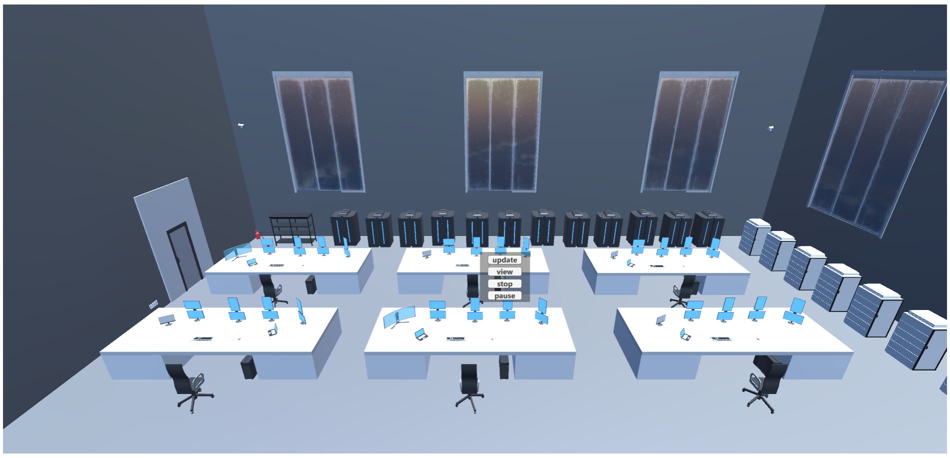

Figure 9. Virtual test platform interface.

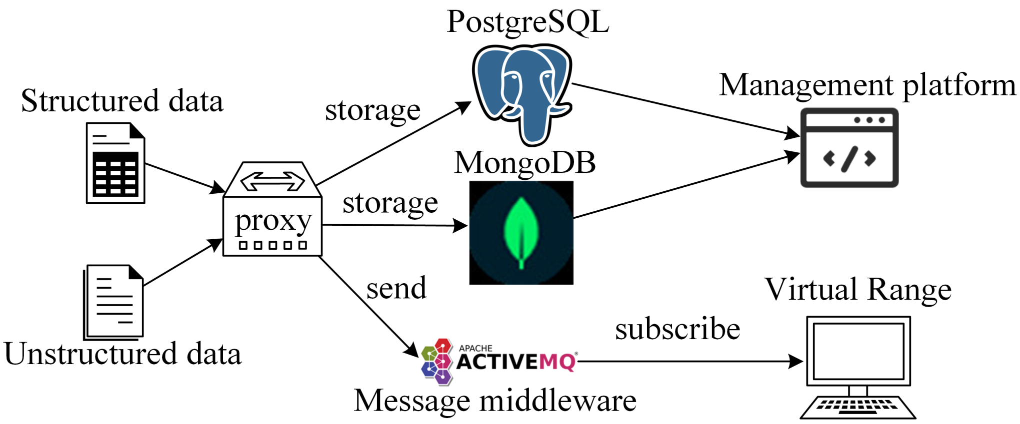

Figure 10. System data flow.

Before the test begins, it is necessary to specify the test plan on the digital twin logical range platform, determine how many cross-regional ranges are involved in the test, and determine how many types of equipment are involved in each range. After starting the test, the digital twin logical range platform can monitor the status of each equipment in real-time and save these data for subsequent analysis work. Taking a flight test as an example, this test involves three ranges distributed in different regions, including two radar simulation equipment, one TSPI simulation equipment, four cameras, and two IMU inertial navigation systems. Figure 11a shows some of the simulation equipment involved in this test, Figure 11b shows the information display interface of the test plan in the digital twin logical range platform, Figure 11c,d show the real-time data reception and display interface of the equipment in the digital twin logical range platform, and the displayed data are synchronized with the equipment in real time. Figure 12 shows the analysis of user behavior data during the test process in the web management platform. Figure 13 shows the interface for viewing the test plan in the web management platform.

For this test, our proposed Logical Range system based on digital twin technology enabled data collection, processing, and analysis across the considered regional ranges. At the same time, the real-time synchronization and display of equipment status information in the real test scene and the virtual twin scene were realized, and the historical test process could be reproduced according to the historical data.

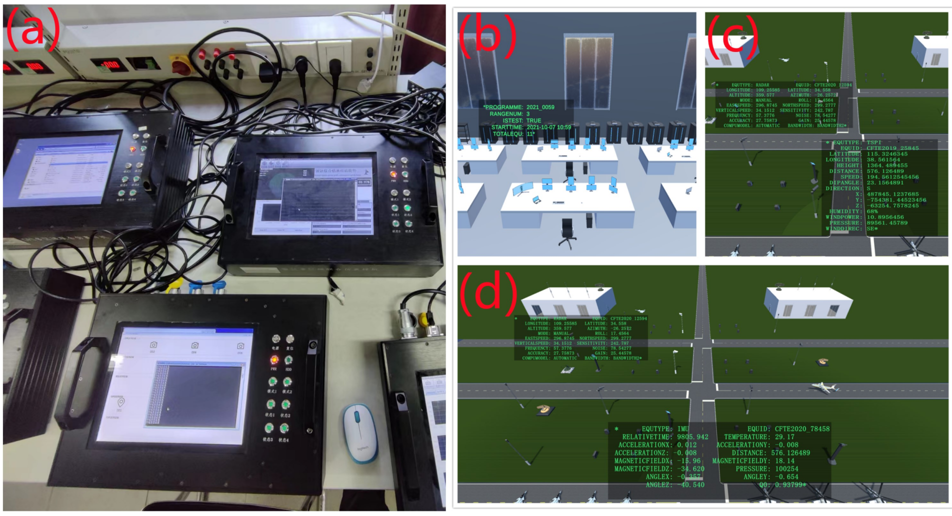

Figure 11. State information synchronous display. (**a**) Some simulation equipment involved in the test. (**b**) Display of test plan information. (**c**,**d**) Equipment data synchronization display.

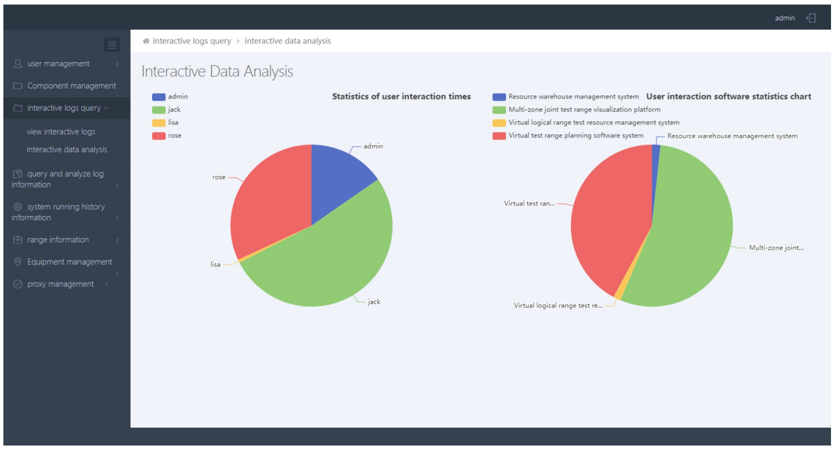

Figure 12. User behavior data during the test process.

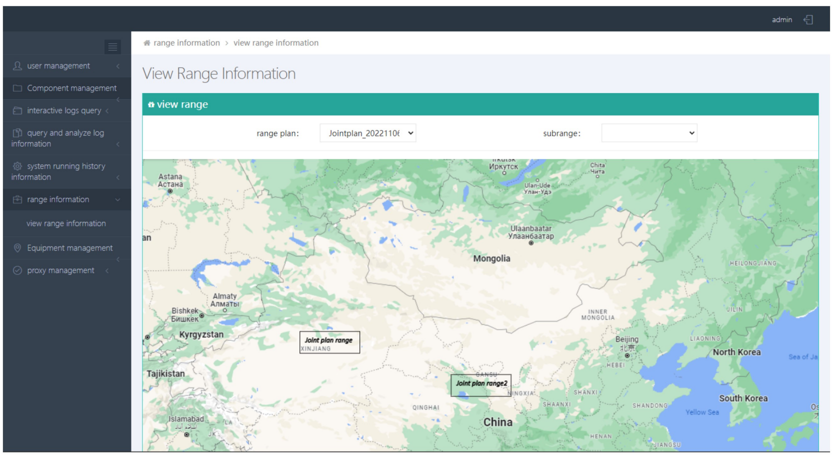

Figure 13. Test plan in the web management platform.

5. Conclusions

Starting from the overall requirements of cross-regional flight testing, this paper analyzes the relevant issues in the current concept of logical range. We have combined digital twin technology with logical range and designed a logical range framework based on digital twin technology, subsequently implementing and validating this framework. At the same time, based on the proposed framework, solutions have been provided to address key issues in the system, such as a proxy-based equipment dynamic extension method, a data communication architecture, and a virtual–real synchronous mapping method. Through testing, it has been shown that our method effectively improves flight test efficiency and digital control level of the test process, while also reducing test costs to a certain extent. Our proposed framework provides key support for the research, application, and development of digital twin technology in virtual test and cross-regional logical range test. In the future, we plan to further focus on digital twin technology and conduct related research on the allocation and scheduling of digital twin flight test tasks.

Author Contributions: Formal analysis, H.C. and Z.D.; Funding acquisition, X.H.; Methodology, H.C.; Project administration, X.H.; Software, H.C.; Supervision, X.H.; Validation, Z.D. and K.W.; Writing—original draft, H.C.; Writing—review and editing, Z.D., X.H. and K.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key R&D Program of China (No. 2022YFB2602203).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We appreciate the financial support of the National Key R&D Program of China, and also thank all the authors of the references that gave us inspiration and help.

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Not applicable.

References

1. Çelik, T.; Gökdoğan, G.F.; Öztürk, K.; Sarikaya, B. An HLA-based tactical environment application framework. *J. Def. Model. Simul.* **2013**, *10*, 217–233. [CrossRef]
2. Noseworthy, J.R. The test and training enabling architecture (TENA) supporting the decentralized development of distributed applications and LVC simulations. In Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, Washington, DC, USA, 27–29 October 2008; pp. 259–268.
3. Guoyu, W.; Runmin, F. Logical Range and joint test training. *Mod. Mil. Aff.* **2006**, *9*, 55–58.
4. Huan, M. Development of Logical Range Planning Furthermore, Verification Tool for Joint Test Platform. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2018.
5. Wei, Y.; Guo, L.; Chen, L.; Zhang, H.; Hu, X.; Zhou, H.; Li, G. Research and implementation of digital twin workshop based on real-time data driven. *Comput. Integr. Manuf. Syst.* **2021**, *27*, 352–363.
6. Bao, Q.; Zhao, G.; Yu, Y.; Dai, S.; Wang, W. The ontology-based modeling and evolution of digital twin for assembly workshop. *Int. J. Adv. Manuf. Technol.* **2021**, *117*, 395–411. [CrossRef]
7. Li, S.; Shu, L.; Wu, G.; Yang, Y.; Yang, M. Digital twin workshop system of circuit breaker based on logic Petri net. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 455–465.
8. Ghaith, M.; Yosri, A.; El-Dakhakhni, W. Synchronization-Enhanced Deep Learning Early Flood Risk Predictions: The Core of Data-Driven City Digital Twins for Climate Resilience Planning. *Water* **2022**, *14*, 3619. [CrossRef]
9. Austin, M.; Delgoshaei, P.; Coelho, M.; Heidarinejad, M. Architecting smart city digital twins: Combined semantic model and machine learning approach. *J. Manag. Eng.* **2020**, *36*, 04020026. [CrossRef]
10. Raes, L.; Michiels, P.; Adolphi, T.; Tampere, C.; Dalianis, A.; McAleer, S.; Kogut, P. DUET: A Framework for Building Interoperable and Trusted Digital Twins of Smart Cities. *IEEE Internet Comput.* **2021**, *26*, 43–50. [CrossRef]
11. Chen, X.; Liu, F.; Ai, Y.; Xu, G.; Chen, J.; Xu, X.; Liang, W. Key characteristics analysis of industrial digital twins for smart manufacturing. *Sci. Technol. Rev.* **2022**, *40*, 45–54.
12. Liu, S.; Lu, Y.; Li, J.; Shen, X.; Sun, X.; Bao, J. A blockchain-based interactive approach between digital twin-based manufacturing systems. *Comput. Ind. Eng.* **2023**, *175*, 108827. [CrossRef]
13. Jwo, J.S.; Lee, C.H.; Lin, C.S. Data twin-driven cyber-physical factory for smart manufacturing. *Sensors* **2022**, *22*, 2821. [CrossRef] [PubMed]
14. Wang, L.; Shu, L.; Zhou, H. Multi-Robot Collaborative Flexible Manufacturing and Digital Twin System Design of Circuit Breakers. *Appl. Sci.* **2023**, *13*, 2721. [CrossRef]

15. Chen, H.; Jeremiah, S.R.; Lee, C.; Park, J.H. A Digital Twin-Based Heuristic Multi-Cooperation Scheduling Framework for Smart Manufacturing in IIoT Environment. *Appl. Sci.* **2023**, *13*, 1440. [CrossRef]
16. Alnowaiser, K.K.; Ahmed, M.A. Digital Twin: Current Research Trends and Future Directions. *Arab. J. Sci. Eng.* **2022**, *48*, 1075–1095. [CrossRef]
17. Liu, S.; Sun, X.; Lu, Y.; Wang, B.; Bao, J.; Guo, G. A Knowledge-Driven Digital Twin Modeling Method for Machining Products Based on Biomimicry. *J. Mech. Eng.* **2021**, *57*, 182–194.
18. Kapteyn, M.G.; Knezevic, D.J.; Huynh, D.; Tran, M.; Willcox, K.E. Data-driven physics-based digital twins via a library of component-based reduced-order models. *Int. J. Numer. Methods Eng.* **2022**, *123*, 2986–3003. [CrossRef]
19. Hu, F.; Yang, Y.; Liu, S.; Zheng, X.; Lv, X.; Bao, J. Digital twin high-fidelity modeling method for spinning forming of aerospace thin-walled parts. *Comput. Integr. Manuf. Syst.* **2020**, *40*, 1–17.
20. Wang, Y.; Liu, Y.; Wang, J.; Zhang, J.; Zhu, X.; Xu, Z. Research on Process Planning Method of Aerospace Engine Bolt Tightening Based on Digital Twin. *Machines* **2022**, *10*, 1048. [CrossRef]
21. Liu, X.; Tang, L.; Zeng, H.; Liu, Y.; Zhang, X. Smart design and simulation of aerospace control system based on digital twin. *J. Syst. Simul.* **2019**, *31*, 377.
22. Liu, Z.; Xia, N.; Du, Q. Application research of spacecraft electrical power system digital accompanying flight technology. *Spacecr. Eng.* **2020**, *29*, 135–141.
23. Liu, Y.; Xie, Q. Technical Characteristics of Digital Twins and Application Prospects in the Field of Flight Testing. *J. Syst. Simul.* **2021**, *33*, 1364.
24. Yan, L.; Quande, L. Design of unmanned aerial vehicle flight simulation platform. *Sci. Technol. Eng.* **2022**, *22*, 12969–12976. [CrossRef]
25. Guo, J.; Hong, H.; Zhong, K.; Liu, X.; Guo, Y. Production management and control method of aerospace manufacturing workshops based on digital twin. *China Mech. Eng.* **2020**, *31*, 808.
26. Jiang, C.; Haibo, W.; Yixiao, Z.; Quanfeng, L. 3D real-time monitoring system of workshop operation for digital twinning. *Mod. Manuf. Eng.* **2023**, *511*, 21. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.