Research on Embedded and Monitoring Test Technology of C^{4}ISR System

Wenyuan Xu[1], Wenjin Yang[3], Chunyu Li[2], Shuo Shi[4], Shengxiao Zhang[1](✉), Hao Li[1], and Haoyang Yu[5]

[1] Systems Engineering Research Institute, Beijing 100094, China
asheng2003@qq.com
[2] Naval Radar and Sonar Repair Plant, Qingdao 266001, China
[3] Military Commission of Combat System in Beijing, Beijing 100094, China
[4] Beijing Information Technology Co., Ltd., Beijing 100094, China
[5] North China Electric Power University, Baoding 071003, China

Abstract. After the C^{4}ISR system is delivered to the user, it is inspected during the equipment support maintenance process that the equipment has low generality, weak automation, and difficult maintenance. This paper proposes the C^{4}ISR system embedded and monitoring test technology, designed componentization, standardization, Integrated embedded and monitoring test environment system architecture, including system usage patterns and logical structures; detailed description of key technologies such as XML (Extensible Markup Language) interface model adaptive technology, component-based system efficient and flexible integration technology and memory-based database data interaction technology; and an embedded and monitoring environment case and system application flow of a C^{4}ISR system, including embedded test process and monitoring test process. The test environment has been successfully applied in many projects, which can realize the recurrence, positioning, confirmation and verification of interface and process problems encountered in the C^{4}ISR system maintenance guarantee process.

Keywords: Embedded test · Monitoring test · C^{4}ISR · Componentization

1 Introduction

The C^{4}ISR system is the center of networked and informationized operations, and is the "force multiplier" [1]. Its equipment quality will directly affect the outcome of a war. The C^{4}ISR system involves the command and control, intelligence reconnaissance, early warning test, communication and navigation, electronic countermeasures, comprehensive support, and distribution of multiple military resources such as combat personnel, and heterogeneous complex military information systems [2], presenting features such as multiple fields and complex structures [3], versatility, frequent information interaction, and high real-time requirements. After the C^{4}ISR system is delivered, on the one hand, the system involves a wide variety of external systems and platforms, complex interactive links, and a variety of interface information. There are many types, many ways of link interaction, and the test equipment is not universal,

Y. Tang et al. (Eds.): HCC 2018, LNCS 11354, pp. 86–92, 2019.
https://doi.org/10.1007/978-3-030-15127-0_9

which makes the system maintenance work difficult. On the other hand, with the increase of the training frequency of the C^4ISR system, the equipment usage rate is high, but the test equipment is inconvenient to cause the system maintenance. In the end, the demand for maintenance tasks is urgent, some factories have heavy repair tasks, system maintenance personnel lack experience, lack of maintenance equipment or backward technology, and the degree of automation is not strong enough to effectively guarantee the normal use of equipment.

Therefore, this article designed a componentized, cross-platform C^4ISR system simulation test environment system framework from how to quickly build C^4ISR system test environment, shorten the development cycle [4]; how to improve the generalization and automation of simulation test, improve the test level of C^4ISR system, realize the recurrence, positioning, validation, verification of interface and process problems encountered in the C^4ISR system maintenance guarantee process, and developed a typical C^4ISR system test environment suitable for a unit, the environment has been successfully applied in many projects.

2 C^4ISR System Test Environment Architecture Design

2.1 Using Pattern Design

According to the maintenance needs of the C^4ISR system, two modes of embedded and monitoring maintenance modes are provided, including.

Monitoring Test Mode. Accessing the test device in the C^4ISR system network, recording the information exchanged by the system, and performing structured analysis on the recorded information based on the standard system message protocol, and assisting the system maintenance personnel analyze the reason of C^4ISR system interface failure.

Embedded Test Mode. By simulating the interface and process of the system associated with the C^4ISR system, assisting the system ensures maintenance personnel analyze the C^4ISR system interface and process failure causes.

2.2 System Logic Structure Design

The architecture of the C^4ISR system test environment [5] should follow the following principles:

a. The test environment should adopt a componentized development method to assemble and independently maintain each module, so as to quickly build a test environment according to the test requirements, facilitate the reuse of modules, improve the sharing and inspection environment, research and development, maintenance efficiency, and reduce development and maintenance costs;
b. The interface test data supports one-time editing and multiple use, providing one-button interface test to improve maintenance efficiency;
c. The test environment should record, store, manage and query the test problems in a unified manner to facilitate knowledge precipitation and sharing;

d. The test environment should be portable, support Windows and domestic operating systems (such as the Kirin) and other operating systems to adapt to various test requirements and improve R&D efficiency;
e. The test hardware environment must be portable and adapt to the use of different spaces;
f. The architecture design should not only make full use of the existing achievements, but also take into account the development needs of future test technologies.

Combined with the C^4ISR system test requirements, based on the component architecture design method, a tailorable interface test environment architecture for multi-professional fields is established, as shown in Fig. 1.

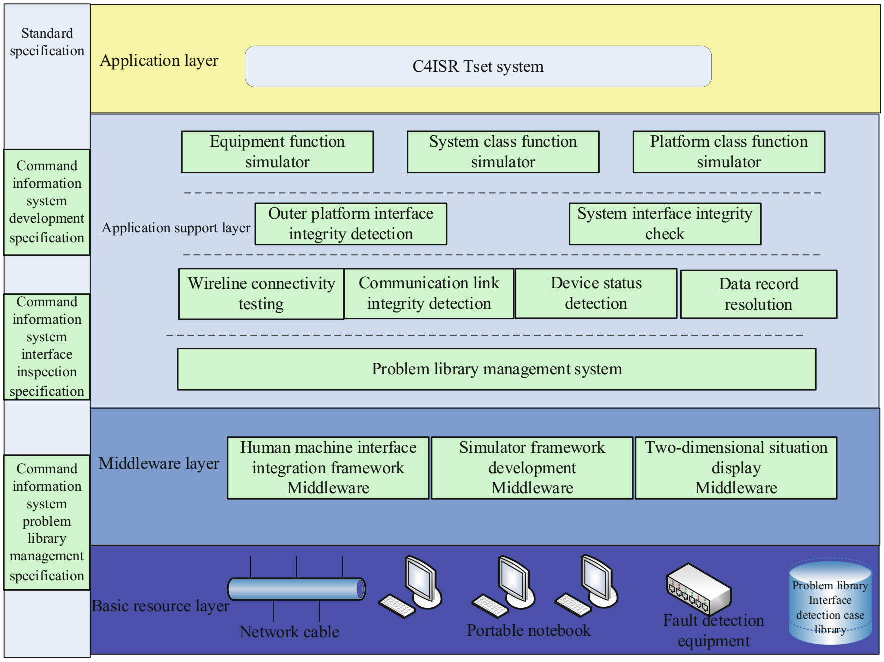

Fig. 1. C4ISR system test environment logic architecture

The architecture consists of four levels, from bottom to top, the base layer, the middleware layer, the application support layer, and the application layer among them:

- The base layer provides an operating environment, including a network cable, a portable notebook, a fault test device, a problem library/interface test case library etc.
- The middleware layer includes a human-machine interface integration framework middleware, a simulator framework development middleware, and a two-dimensional situation display middleware.
- Application support layer consists of equipment function simulator, system function simulator, platform function simulator, external platform interface integrity test,

system interface integrity test, wired connectivity test, communication link integrity test, equipment state inspection, data collection analysis, and problem library management systems. The equipment function simulator provides functions such as navigation and radar [2, 6]; the system function simulator provides the associated command system functions; the platform function simulator provides the external platform system functions; and the wired connectivity test provides wired network connectivity test; The problem library management system provides problem libraries, case additions, queries, modifications, and deletions.

- The application layer is mainly based on the test target, and the human-machine interface integration framework is used to assemble to form the C^4ISR test system.

3 Key Technologies Implemented by the System

3.1 XML-Based (Extensible Markup Language) Interface Model Adaptive Technology

An important task of C^4ISR system simulation test is the match of the message interface. However, due to different equipment and different development time of the same equipment, the format of the operational protocol is different. If the message is docked through hard code, different equipment and interface docking code are different, resulting in code duplication of development, multiple versions of maintenance difficult and inefficient. To this end, an interface method for combat protocol interface based on XML [7] is proposed, which shields the relationship between code and specific message format, and realizes the change requirements of the protocol through the configuration file. Among them, an example of the combat message abstract model is as follows:

```
<Protocol Name=”TimeMessage>
<Element name=”NUMBER“,   Type=” unsigned int“,   Length=”1“/>
<Element name=”ID“,   Type=” unsigned int “,   ValidValue=“128”,   Length=”1“/>
<Element name=”LENGTH“,   Type=” unsigned int “,
ValidValue=“88”Length=”2“/>
<Element name=”TIME“,   Type=” unsigned int “,   Length=”4“>
<BasicInfo LSB=“0.1” Unit=“MS”/>
</Element>
<Element name=”Control1”,Type=”CONTROL” Length=“4”>
<subelement name=”spair”,   Begin=“0”,   End=“0” Type=“spair”/>
<subelement name=”OperationType”,    Begin =“1”,   End=“2” Type=“Code”>
<EncodeItem Type =“COMMON“,   Value=”0“ Content=“one”/>
<EncodeItem Type =“COMMON “,   Value=”1“Content =”two”/>
<EncodeItem Type =“COMMON “,   Value=”4“ Content=“custom”/>
</subElement>
</Element>
</Procotol>
```

3.2 Component-Based System Efficient and Flexible Integration Technology

The test environment function module includes monitoring test and embedded simulation functions. The functions are complex. The system software needs to be designed and integrated by layered and componentized architecture technology. The mode of flexible organization forms different system functions. Firstly, according to the test requirements, the test function is divided into network class and business class interface plug-ins. The plug-in function is independently developed according to the plug-in interface development specification provided by the human-machine interface. The provided interfaces include plug-in loading, plug-in uninstallation, plug-in running/stopping, plug-in initialization, plug-in reverse-initialize, etc. then, according to the test object, the function to be tested is configured; finally, according to the configured test function, each plug-in is started to provide a test system for the system maintenance personnel.

3.3 Data Interaction Technology Based on In-Memory Database

The application layer communication protocol refers to a format for transmitting messages between application processes on different end systems defined by a program developer. The data in the system is transmitted on the basis of the application protocol. The data interaction technology based on the in-memory database can automatically resolve the application layer communication protocol and provide a data interaction cache based on the in-memory database. With the dynamic parsing function of the message format, the user dynamically defined message format can be automatically parsed into an application. The data parsing and interaction technology based on in-memory database provides a set of efficient and reliable management mechanism for experimental verification system data analysis and real-time data storage.

4 C^{4}ISR System Verification Environment Application Case

A C^{4}ISR system provides embedded and monitoring analog fault test, while providing network line test, functional and physical connections to the system as shown in Figs. 2 and 3.

(1) In the case of monitoring test, the system is connected to the C4ISR system network, the integrated test device records the interface information of the C4ISR system interaction, and based on the standard C4ISR system message protocol, the recorded information is structured and analyzed;
(2) In the embedded simulation, the integrated test equipment simulates the equipment system, the operational command system and the platform type interface, and performs interface check with the C^{4}ISR system.

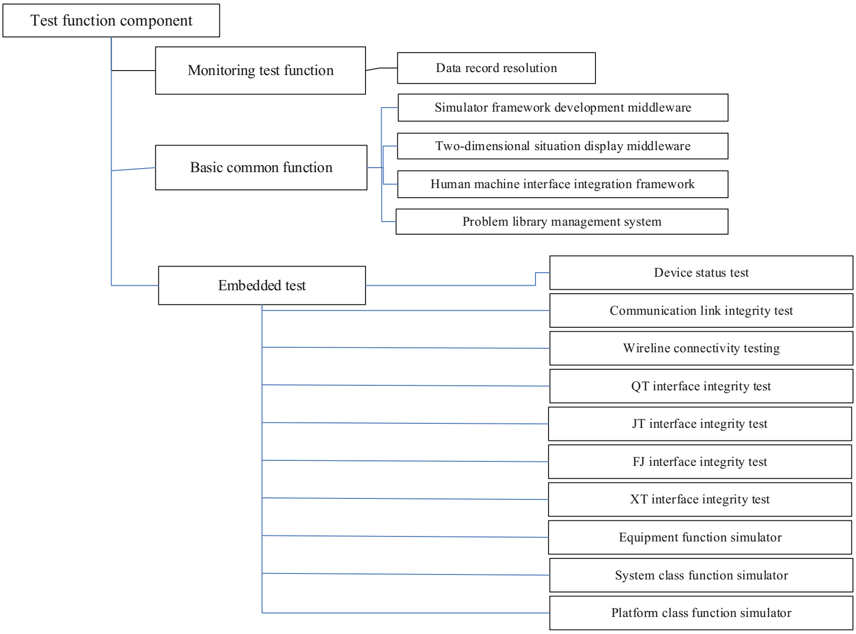

Fig. 2. C4ISR system function chart

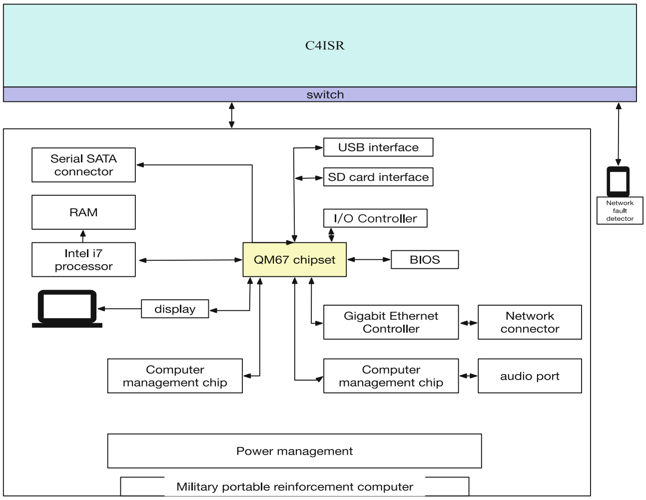

Fig. 3. C4ISR system test environment physical connection diagram

5 Conclusion

In accordance with the requirements of modularization, serialization and standardization of test equipment, this paper studies the test equipment. According to the principle of "uniform design, parallel development and comprehensive integration", based on the modular and component design concept, adopts the unified design idea and follows the stable application framework design, relies on standard information protocol and link channel mode to realize the design research of C^4ISR system test equipment, achieve standardization and general design of test equipment, and develop a set of test equipment for C^4ISR system for fast and convenient to solve the interface and process problems encountered in the C^4ISR system during the maintenance guarantee process, and reproduce, locate, confirm and verify the problem. It plays an important role in system joint debugging, interface docking, and function verification, and can improve the maintenance support capability of the factory for the C^4ISR system. However, as the maintenance capability is further improved, the interface and process test are difficult to meet the requirements. It is necessary to add the one-button performance test and other test functions on the existing basis, so as to realize the rapid improvement of the inspection and maintenance capability and effectively ensure the use of the equipment.

References

1. Wang, W., Liu, G., Yang, H., Yang, X.: Theory and process of validation for C4ISR. Command Control Simul. **36**(06), 88–91 (2014)
2. Peng, H., Wang, P., Zhang, X.: Research of C^4ISR system warfare simulation test technology in battlefield environment. Fire Control Command Control **38**(8), 158–168 (2013)
3. Fu, J., Wang, J., Zhang, S.: Software testing method for integrated electronic information system. Command Inf. Syst. Technol. **6**(01), 87–92 (2015)
4. Sun, X., Fu, J., Lu, L., Yang, X.: Simulation testing method for integrated electronic information system software. Command Inf. Syst. Technol. **5**(4), 75–79 (2014)
5. Fang, L., Chen, H., Shao, D.: Analysis of software simulation test system construction. Ship Electron. Eng. **32**(6), 12–16 (2012)
6. Peng, J., Shu, C., Zhang, G., Chen, S.: Research on simulative test of one C^4ISR system. Fire Control Command Control **34**(S1), 169–171 (2009)
7. Fan, W.: Is computer simulation independent life? **21**(24), 7982–7984 (2009)