Reliability Design for Data Storage in an Integrated Test System

Zhang Bing¹, Sun Xiao¹, Shi Jianwei¹ and Deng Rongjun²

¹China Astronaut Research and Training Centre, Beijing 100094, China;
²Beijing Xuan Yu Space Technology Company, Beijing 100039, China

Email: gfkdzb@163.com

Abstract. To solve the failing errors of prompting message and storing files which are occurred in a physical-files-receiving configuration software item installed on an Integrated Test System, some specific measures have been taken. The receiving-file software item is upgraded and tested. The upgraded item is validated by many different examples, and the results show that the reliability of the item is significantly improved.

1. Introduction
The Integrated Test System is a testbed for different communication circumstances. It can receive and store files and data from spacecraft, and make an automatic classification, display, storage and inquisition for different categories of data[1-2].

The system is based on multi-platforms for Spacecraft Integrated Test and data management projects. It is an open-up and modularized structure with a stable and coincident software framework. The system can be reconfigured and rebuild with the reconfigurable management tools for the front and the rear user interfaces. The system has open-ended interfaces for the future mission with the general and standardized inner modules.

The system can be flexibly deployed in multi operating system (Linux and Windows) without running errors with other applications and software. Even under the fault or unreasonable input circumstance, it also has a higher anti-interference and anti-malfunction ability. With the robust structure, it can give many prompts for the clients without mistaken-running.

The data flow-chart and software configuration schematics are illustrated in Figure1.
The Interface software (CDI/SDI/C3I) receives the data from the tele-linking network, and will identify the source and categories of the data according to the fixed head-bytes. The raw data will be transferred to the Data Receiving and Processing (DRP) software.

The function of DRP software is to manage inner data transferring and decomposing. First it will store the raw data after receiving them from the Interface software. With the data protocol configuration tools and data configuration management tools, it will get the specific protocol and processing methods in the form of the configurating files. Secondly, the DRP software gets the protocol and the predefined methods in the classification, transformation, extraction and decomposition for the data and uses them to decompose and process the data. Finally, the results of the processed data will be sent to the other software, including the Classification Server software, Monitoring and Displaying software, Physical Files Receiving software, Stream Media Receiving software, Data Storing software.

The Monitoring and Displaying software start with the different managerial authentication, and will display the different kinds of data with original number, the waveforms of the physiology data. The Physical Files receiving software and Stream Media Receiving software will accept the data and make the final data files in the form of byte streams. The data files will be stored in the disk array of the data base, and the index will be established in the database.

The Classification Server software will interpret the decomposed data according the knowledge base. The interpreting results will be sent to the Classification Client software.

The function of the Data Storing Software is to store and manage all the decomposed data. The Data Inquiring software will inquire the raw data and the decomposed data under the inquiring conditions, and display the inquiring results with the displaying requirement.

The low orbit spacecraft fly around the Earth in 90 minutes. The ground system can receive data only in the observing and controlling zone. If the received data is incomplete, it will send a requirement for resending the data. Due to the limited time in the observing and controlling zone, if the data and files are received and stored successfully, but the ground system make a wrong requirement for resending the data only for the unsuccessful prompting message.

Figure1. The data flow-chart and software configuration schematics
communicating resources will be wasted, thus the reliability for receiving and storing data in the ground system is an important factor[3].

2. The problem of the traditional design
The Figure 2 is the flow chart based on the interface, function and performance requirements.

![Flowchart of data receiving and storing](image)

**Figure2.** The Receiving and Storing Data Flowchart

The process is described as below.

a. The raw data from the tele-linkage to the interface server, include: head, body and tail of the data;
b. Storing source codes after getting them, forward them to the file-receiving server.
c. Buffering the source codes in sequences after receiving them, if some packets are missing, ask for resending by a reply token.
d. Processing the buffer data until the tail of the data, send the integrity message back to the receiving server, then reload by it. If the data is incomplete, then send the failure token.
e. The receiving files are storing in the buffer memory
f. Send the files to the analysis server through FTP, and store the records in the server, mark the success message after storing files.

The procedure for recording the files in receiving software item is:

a. Verify the connection with the data base when starting up the receiving process. A connecting success message with the server will show up in case of losing any data from database.
b. The connection will be keeping through the whole process, waiting for the result.
c. Once the data is successfully stored, *Receiving Files Successfully* will prompt on the interface of the software. The connection will still be kept.

After the data is completely received, it will often find some problems:
a. The source codes are received successfully and the data file is constructed well, FTP is good, but the record is unsuccessfully written to the database, that means even after receiving files, Receiving Files Successfully message is still not prompted.

b. The source codes are not received successfully, the files are bad, and Receiving Files Successfully is still not prompted.

After checking the system’s log files, the first problem is caused by the failure of writing records to the database because of the unstable connection with the database. Once the connection is built up, it will not break with it actively and reconnect it next time. The failure for writing to database is due to the unstable or abnormal connection.

The second problem is caused by the failure of the buffer hardware in the receiving module. Since the bad sectors can be existed, and the uncompleted source codes will make the files unsuccessfully done.

Above all the two problems, the source codes are all successfully received, but no message is shown up and no follow-up methods are available. The requirements for resend the files will waste the precious tele-linkage resources between the spacecraft and the ground control system.

3. The Improvement Measures
To make sure the whole process is completed successfully from receiving source codes, uploading to FTP server and writing, three methods are proposed.

3.1. The prompting message method
The four steps for receiving, creating, uploading to FTP, and writing to the database are divided for the process of receiving and recording the physical files. The prompting message is only showed on the steps of receiving source codes and writing to the database successfully.

Now the prompting message will be showed on every step, they will help to monitor the whole process. Once the uploading to FTP is successful, it will not require to send the source codes again.

3.2. The fault-tolerant method
Using the fault-tolerant method[4], it will successfully complete the whole process even either of the creating, uploading and recording process is failed.

3.2.1. The software fault-tolerant technique and recovery block. The software tolerant technique is a redundant method for the mistakes in the software. These mistakes can be found or miss-found. But the fault-tolerant technique can provide a mechanism instead of falling into a failure. It includes Recovery Blocks(RB), N-Version Programming(NVP), Consensus Recovery Block(CRB), Acceptance Voting(AV) and N-Self-Checking(NSC).

The strategies include: Forward Recovery and Backward Recovery. Forward Recovery is to continue the current computing process to make it consistently in a right way even the current state is incorrect. Backward Recovery is to return back to the previous right state. As is showed in Figure 3.
The *Recovery Block* is to choose a group of blocks as fault-tolerant blocks, changing them from normal program blocks into *Recovery Blocks*. The blocks can be software module, sub-program, program segments or procedures. One *Recovery Block* include many blocks with the similar function and different design. Only one block is running at a specific moment. The dynamic redundancy is made by replacing a failure one by the backup *Recovery Block* while the whole system is keep running properly.

### 3.2.2 Increasing the recovery button for the receiving and storing procedure

Three recovery buttons are added for the receiving, creating and FTP uploading files procedures. Once the time is over 15 minutes and there is no message for the next step, the recovery buttons can be pressed to implement the process due to the file capacity is no more than 1Gigabytes. These buttons can make sure that the above three processes are successful completed.

### 3.3. The modified connection mode for the receiving software and the database

The failure is caused by the unstable connection between the receiving software and the database. To solve this problem, the connection should be confirmed before writing the records to the database. Add the corresponding software codes for the confirmation of the connection, and reconnect the database until it is normal. In this way the connection is constantly working well before any actions.

### 4. Validation for The Improvement Measures

#### 4.1. The improved connection processes

Adding the software codes for the connection with the database, it will reconnect to the database until it is normal.

In a simulated testing circumstance, the transmitting process is normal even the connection is compulsory disconnected.

#### 4.2. The improved message prompting process

The following optimized steps are applied for the receiving source codes, creating files, uploading to FTP, and writing records to the database:

a. Normal testing: updating the software, simulating the procedure of the tele-linkage of the spacecraft, implementing several times, the results are normal.

b. Simulated loss packets testing: transmitting a large file with the capacity over 1.5G. In order to simulate the loss packets, the cable line is pulled out. The error message is fed back to the linkage of the communication network. The data will be re-sent until they are successfully received and stored.
4.3. Adding the recovery buttons

There will be a prompt message for over 15 minutes since the capacity of the file is less than 1G. If this happened, press the recovery button.

a. Simulated test for the failure of creating and uploading files to the FTP server: The test is simulating the procedure of creating and uploading files to the FTP server with a large file (over 0.5G). While the message for the source code is normal, pull out the cable line for the receiving software, the cable line is unconnected with the FTP server. Install the cable line again, press recovery button, the files will be re-created in the receiving software item, re-uploading to the FTP server. The procedure will insure the whole process is completed normally.

b. Simulated test for the failure of writing to the database[5-6]: After the source codes are completely received, the prompt message is shown up. Stop the service of Oracle in database using the command Shutdown Immediate. Then start the database with SQL command Start up, press recovery button, the files will be processing again by the receiving software item, the results will be sent to the database successfully.

The above two tests can show that the files can be processed properly even the database is abnormal.

5. Conclusion

The Integrated Test System is optimized with three proposed measures including the prompting message method, the fault-tolerant method, the modified connection mode for the receiving software and the database. The three proposed methods are validated with many examples. It has shown that the reliability of the system is improved, while the precious communication resources are preserved.

Reference

[1] Hua Mao Wang, Jindong Yan, 2017. Electrical Test Technology of Spacecraft. (China: National Defense Industry press, Beijing)
[2] Hua Mao Wang, 2018. Spacecraft electrical test technology. (China: Beijing Institute of Technology press, Beijing)
[3] Gary Musgrave Ph. D, Axel Larsen, Tommaso Sgobba, 2011. Safety Design for Space Systems. (UK: Elsevier Limited of the Boulevard, Langford lane, Kidlington, Oxford)
[4] Lorenzo Strigini, 2004. Fault Tolerance Against Design Faults. Available online.
[5] Saikat Basak, 2010. Oracle DBA Concise Handbook Covers 9i to 11g. Available online.
[6] http://docs.oracle.com/cd/E11882_01/appdev.112/e13995/oracle/jdbc/OracleConnection.html.