



Data Management Software System Design for Spacelab

Panpan Zhan^(✉), Lan Lu, Xiongwen He, Luyuan Wang,
and Yong Sun

Beijing Institute of Spacecraft System Engineering,
Beijing 100094, People's Republic of China
panpan3210@qq.com

Abstract. The technology of propellant refueling is one of the important technologies of spacelab and cargo ship rendezvous and docking mission. With the data management software system, the function of two aircraft docking bus network is established, and the function of data transmission and control is realized. In the spacelab task, a data management software system for propellant refueling of cargo spaceship was proposed. Firstly, the software requirements and design principles are analyzed. Then, the layered software architecture is designed, and the rendezvous and docking task is expatiated. Finally, the realization of propellant refueling process in spacelab and cargo ship is illustrated. The research results can provide an important basis for the design of the data management system for the propellant refueling on space station.

Keywords: Spacelab · Data management · Rendezvous and docking · Propellant refueling

1 Introduction

According to China's manned space engineering development strategy [1], the space laboratory's main task is to verify the rendezvous and docking with various transport spacecraft, and to verify the space laboratory's onboard propellant refueling. It accumulates experience for the space station. In addition to supporting manned spacecraft rendezvous and docking missions, one of the important tasks of the space laboratory is to realize onboard propellant refueling and space-to-ground or ground-to-space transmission of goods with cargo ships.

Data management system is an important platform system in space laboratory. It mainly realizes the functions of telecommand, telemetry, bus network managements, flight program control, flight state management and rendezvous and docking mission. In the process of rendezvous and docking between space laboratory and transport spaceships (manned spaceships or cargo spaceships), the data management system is responsible for establishing the docking bus network between the two spacecraft. It realizes the functions of telecommand, telemetry acquisition and control data bidirectional transmission for the two spacecraft. The data management software system completes the above functions with the hardware cooperation, and realizes the data

transmission and control of the propellant refueling data during the rendezvous and docking process with the cargo ship.

The data management software system supporting the propellant refueling with cargo ships is an important control platform of the onboard refueling system. At present, only ISS [2, 3] has applied the space propellant refueling system.

Based on the requirements of rendezvous and docking mission, this paper designs a data management software system for space laboratory, which can support multiple rendezvous and docking modes. It supports propellant refueling functions with cargo ships. The research results in this paper can provide reference for the mission of china's space station several years later.

2 Requirements Analysis and Design Principles

The data management software of the spacelab mainly has the following requirements:

- (1) Telemetry management. It includes the functions of telemetry data acquisition, telemetry source packets organization, source packets scheduling, AOS [4] frames generation and downlink.
- (2) Telecommand management. It includes the functions of receiving the telecommand data, checking and verifying the format, processing the real-time instructions and forwarding the telecommand data to other applications.
- (3) Bus network management. The data management software realizes the functions of data acquisition, instruction sending and data interaction of all devices on the whole satellite bus. The docking 1553B bus is used to realize the data communication function of the spacelab and the cargo ship in the mode of rendezvous and docking. It realizes the data interaction and control function between the spacelab and the cargo ship.
- (4) Flight status management. It can identify and control the key contact signals (launch takeoff, separation of vehicle and rocket, emergency shutdown of carrier) and docking signal interfaces (docking and separation signals) of the whole satellite. It realizes the conversion of flight status from ground flying stage to ascending stage and the conversion from ascending stage to normal flying stage. It performs corresponding flight programs in the flight status conversion.
- (5) Rendezvous and docking management. It realizes the rendezvous and docking functions with manned ships or cargo ships. It can support the conversion of normal operation mode, docking mode with manned ships or cargo ships. It realizes the telemetry data, telecommand data, propellant refueling data and other control information routing and transmitting after the docking of the two spacecraft and their buses connected with each other.

In view of the complexity of the upper task and the multi-layer of the underlying bus network in the data management software of the spacelab, the software system design is based on the layered method. Combining with the idea of component design, it is organized into a number of software tasks to drive the software's operation. The Spacecraft Onboard Interface Service (SOIS) [5] of the Consultative Committee for Space Data Systems (CCSDS) and the Packet Utilization Standard (PUS) [6] of ESA

are fully used in the design of the hierarchy, components and tasks. The main design ideas are as follows:

- (1) Design of low level coupling architecture. It aggregated the application tasks, functional modules and underlying drivers to form a single-body system in existing data management software. Errors in one module resulted in errors in other modules. Therefore, the excessive coupling of each module leads to reliability degradation. Data management software of spacelab adopts hierarchical architecture design. Operating system and hardware driver are placed in the bottom layer, and the top layer is the tasks of user application layer. The data management software is divided into application layer, middleware layer and driver layer. The driver layer realizes the driver of each hardware interface. The middleware layer completes the specific functions of each business module. The application layer calls the business module of the middleware layer to realize the task's functions. Hierarchical low-coupling architecture can reduce the coupling between system modules, shield the impact of the replacement of hardware interface and protocol on the upper software, and improve the flexibility and reliability of the system.
- (2) Component design. Onboard software component [7] (component for short) refers to a functional module that can be released independently and assembled by a third party. It meets certain spacecraft software requirements and provides services through interfaces. It should have two basic characters. Firstly, component users can assemble components according to established rules. Secondly, it must have specific, one or more functions. It can be called by the user, and run correctly in a predetermined operating environment. In order to improve the universality and adaptability of onboard software, and improve the efficiency and reliability of software development, this paper uses component-based design of transport layer, application support layer and system business layer components to accomplish business function encapsulation and module design.
- (3) Multiple flight modes are independent from the rendezvous and docking task. During the flight mode conversion, different exchange and docking management tasks are performed according to the different spacecraft. The functions of independent operation mode, spaceship data management function in docking mode with manned ship, and propellant refueling after docking with cargo ship are realized respectively. Since the flight mode switching states are complex, it is more complex for the rendezvous and docking task to intertwine with it. By separating various flight modes and the rendezvous and docking task independently, the rendezvous and docking task in different docking modes can be realized, which makes the manned ships data management function and cargo ships propellant refueling function independent and definite.

3 Software Architecture Design

In order to improve the reliability and flexibility of the data management software system and to support multiple rendezvous and docking functions such as cargo ship's propellant refueling management, this paper designs the data management of spacelab

according to the principles of hierarchical low-coupling architecture and component-based design, and separation of the multiple flight modes and the rendezvous and docking task. The architecture of the spacelab's software system is divided into three layers: driver layer, middleware layer and application layer, as shown in Fig. 1. The middleware layer is divided into transport layer, application support layer and system business layer.

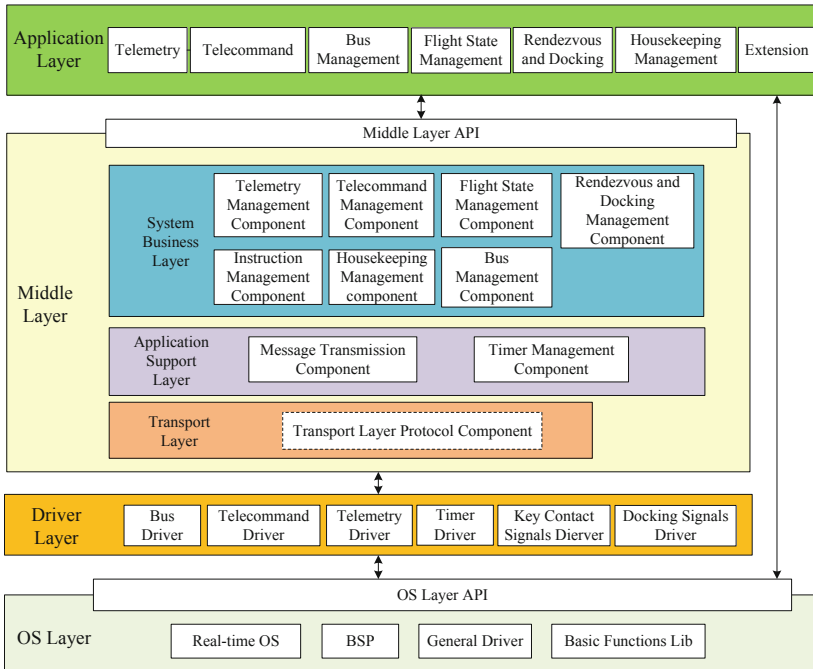


Fig. 1. Data management software architecture of spacelab

3.1 Driver Layer

In order to reduce the dependence of the software on the hardware changes, hardware-related interfaces are encapsulated into hardware drivers, including bus driver, remote control driver, telemetry driver, clock driver, key contact signals driver and docking signals driver. These drivers together with the components of the middleware layer realize the corresponding functions. Driver layer provides hardware-specific drivers for the system, which can shield the differences between hardware platforms and effectively ensure the independence and portability of the middleware layer and application layer.

3.2 Middleware Layer

The middleware layer is a general middle system located between the driver layer and the user application layer. It completes the specific functions of each business module.

It provides standard program interfaces in the form of onboard components. The upper application software can obtain business services independently through the standard business interfaces provided by the middleware. In order to make the middleware have good expansibility and maintainability, the middleware is divided into three layers.

- (1) Transport layer. It provides standard interfaces for the upper services. The transport layer protocol implements the 1553B bus transmission protocol and the docking bus transmission protocol.
- (2) Application support layer. It includes message transmission component and time management component, providing inter-task communication and unified time management function for data management system respectively.
- (3) System business layer. This layer focuses on the system businesses related to data management software functions. The business component includes the following components.

Telemetry management component. It realizes the telemetry data acquisition, telemetry source packet organization, source packet scheduling, AOS frame organization and downlink functions of each systems of the spacelab.

Telecommand management component. It realizes the functions of receiving and checking the telecommand data, routing the telecommand data, unpacking and executing the telecommand data.

Instruction management component. It realizes the real-time instruction execution, delay instruction execution, instruction routing and recording functions.

Flight state management component. It realizes the processing of flight state conversion, and executes corresponding flight procedures, which can update and maintain onboard flight procedures.

Rendezvous and docking management component. The Rendezvous and docking function of the data management system is encapsulated. According to the different modes of the visiting spacecraft, the module can support the bidirectional transmission and management of propellant refueling information after docking with the cargo ship.

Bus management component. The application data of the upper layer is divided into message frames conforming to 1553B bus protocol and sent to the transport layer for processing.

Housekeeping management component. It realizes the functions of housekeeping management and autonomous management of the spacelab, including memory management, onboard maintenance, event reporting and onboard monitoring.

3.3 Application Layer

Application layer is composed of many tasks (or processes). These tasks use the standard interface provided by middleware layer for task-related processing, to complete the tasks specified by the spacelab.

3.4 Rendezvous and Docking Task Design

The manned ship and cargo ship carry out the rendezvous and docking mission with the spacelab respectively. The rendezvous and docking process with the cargo ship is a new and key verification activity, which can realize the important functions of onboard propellant refueling and space-to-ground or ground-to-space transmission of goods. Therefore, the design of rendezvous and docking mission is an important part of the data management software system of spacelab, which can realize the bus interconnection, telecommand, telemetry downlink and data transmission control after docking with spacecraft.

According to the design principles of separation of multiple flight modes and rendezvous and docking task, the rendezvous and docking task is not started when spacelab is flying independently. In this case, the rendezvous and docking task is not performed. The rendezvous and docking task is only started when the flight mode is changed to docking mode with a manned ship or a cargo ship. The rendezvous and docking task completes the rendezvous and docking function with the execution of other tasks. The communication method between these tasks is shown in Fig. 2.

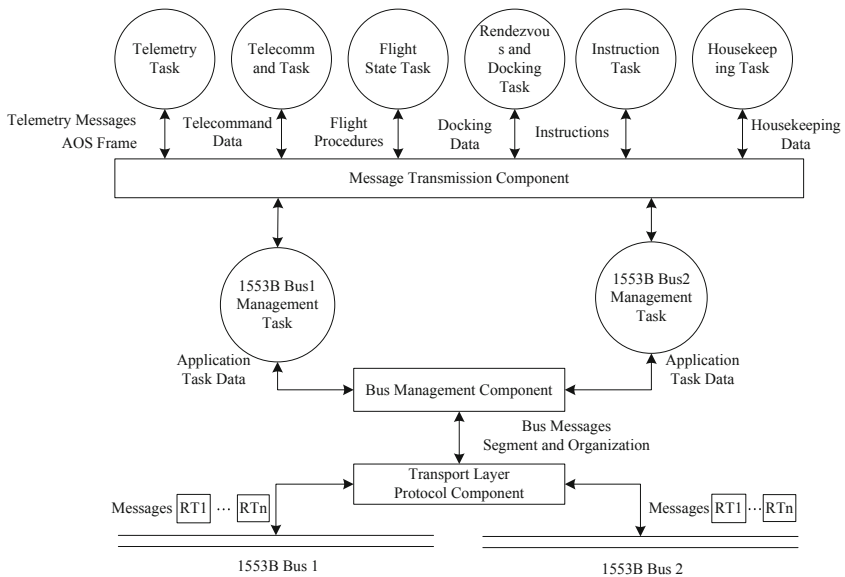


Fig. 2. Interaction method between multi-tasks

From the view of telecommand data stream and telemetry data stream, the communication process between each task in data management software is explained.

Telecommand data flow design. After receiving telecommand data from ground, telecommand driver sends them to telecommand task through message transmission component. Telecommand task invokes telecommand management component to check and resolve the telecommand data. When the format is correct, if they are the

data of other devices on the bus, the telecommand management component sends the data to the remote terminal through the bus. The bus management component and transport layer protocol component are invoked and finally forwarded to the corresponding device through 1553B bus. If the data are sending to the software itself, they are forwarded to other tasks through message transmission component.

Telemetry data stream design: The telemetry data stream is divided into the telemetry data collected from bus and the telemetry data collected by software itself.

The process of the telemetry data collected from bus is that the telemetry task registers the data to be collected to 1553B bus management task, and starts the periodic collection. The 1553B bus management task organizes the bus messages of RT -> BC through the bus management component and the transport layer protocol component, and then starts the task. After the bus messages return, the bus management component sends the message to the telemetry task. The telemetry task will organize telemetry source packet from telemetry data collected on the bus.

The process of telemetry data processing of the software itself is that the telemetry task periodically sends a notification message to each task through the message transmission component. The other tasks send their own telemetry data to the telemetry task after receiving the notification message, and the telemetry task organizes them into a software telemetry source packages. And then, the telemetry source packages of other devices and the software's own source packets are organized into AOS frames. Finally, the AOS frames are sending to ground.

The process of rendezvous and docking management task is shown in Fig. 2. The task runs periodically. Firstly, the telecommand data sent from the telecommand task is processed, the telecommand data is sent to the cargo ship through the bus management task. Then the control command is processed to control the process of propellant refueling. If the propellant refueling process is started, the data of propellant refueling are collected from the propellant refueling manager on the 1553B bus and forwarded to the cargo ship. If it is the docking mode with the cargo ship, the service request for data communication is queried, and the communication data status with the cargo ship is set according to the service requests. The service requests include the cargo ship real-time telemetry service request, delay telemetry service request, advance supplementary management data service request and display parameter service request, etc. If there is a service request, the data communication state is set up. It generates a set of RT -> BC 1553B bus messages, receives the propellant refueling data and telemetry data from the cargo ship, and sends them to the propellant refueling manager in spacelab.

4 Rendezvous and Docking Communication Process Design

The communication process of rendezvous and docking is supported by the data management software system of spacelab and cargo ship after the connection and interconnection of the two 1553B buses. It controls the action of the propellant refueling devices, such as the liquid floating device and compressor, to connect and close the gas-liquid pipeline of the propellant refueling system, so as to realize the propellant refueling process.

After rendezvous and docking, the communication network between the cargo ship and the spacelab establishes a physical connection based on the docking 1553B bus. Under the control of the data management software of the spacelab, the functions of telemetry acquisition, data frames organization, telecommand transmission and propellant refueling of the cargo ship are completed. CTU is the data management center computer and the physical equipment in which data management software is running. Under the control of software, the two aircraft communicate with each other by transmitting the state information which represents the state of their propellant refueling functions.

Before proceeding with the propellant refueling process, the propellant refueling function of the spacelab data management software is started by executing the propellant refueling starting instruction. The software collects the propellant refueling disposal information and the fault detection information of the propellant refueling manager, sends them to the cargo ship by docking 1553B bus. The data management software of the spacelab collects the service request of data communication, inquires the data of propulsion replenishment sent by the cargo ship, and sends it to the propulsion Replenishment Manager to realize the control of the valve and compressor of the propulsion replenishment system of the space laboratory. After the propulsion refueling is completed, the refueling stopping command is sent through the flight program, and the refueling system valve is closed to stop the transmission of the propellant refueling information between the two spacecraft.

5 Conclusion

Onboard propellant refueling is one of the key technologies in manned space station project. Data management software system, as the information bridge of rendezvous and docking between spacelab and cargo ship, realizes the rendezvous and docking of two spacecraft and the function of propellant refueling. This paper presents the architecture of data management software system for space laboratory, designs the mission of rendezvous and docking management, and gives the process of propellant refueling of spacelab and cargo ship. It can realize the pre-verification of onboard propellant refueling technology for cargo ship and space station.

References

1. He, Y., Yang, H., Bai, M.: Spacelab technology summary and development stratagem. *Manned Spacelight* **15**(3), 10–17 (2009)
2. Zimpfer, D., Kachmar, P., Tuohy, S.: Autonomous rendezvous, capture and in-space assembly: past, present and future. In: *AIAA 1st Space Exploration Conference: Continuing the Voyage of Discovery*, Orlando, Florida, pp. 321–328 (2005)
3. *International Space Station Communications Systems*. Washington D.C., USA, pp. 2–17 (2005)
4. Consultative Committee for Space Data System: AOS space data link protocol. CCSDS 732.0-B-2. Recommended Standard, Issue 2, Washington D.C., USA (2006)

5. Consultative Committee for Space Data System: Spacecraft Onboard Interface Services Concepts and Rationale. CCSDS 830.0-G-0.4, Washington D.C., USA (2005)
6. European Cooperation for Space Standardization: Space Engineering: Ground Systems and Operations - Telemetry and Telecommand Packet Utilization. ECSS-E-70-41A (2003)
7. Zhang, Y., Yuan, J., Guo, J.: Design of reconfigurable general TM based on software components. *Spacecraft Eng.* **22**(4), 62–67 (2013)