A Method of Automatic Code Generation for Spacecraft OBDH Software

Hongjun Zhang(✉), Li Pan, and Mengmeng Yu

Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
zhanghongjunbuaa@sina.com

Abstract. The numerous interfaces of spacecraft OBDH software and frequent changes in requirements, resulting in the low efficiency and reliability of the manual coding of OBDH software. An automatic code generation method based on electronic data sheet (EDS) is proposed. The EDS system is introduced, and the output of the EDS system can be used to generate OBDH software code automatically, which improves the efficiency of software development. An structure of OBDH software is designed, which separates the logical code from the parameter code. Due to the EDS system data source is unique, and software code is automatically generated by tools, which avoids the mistakes of coding manually and promotes the reliability of OBDH software and even the reliability of spacecraft is improved.

Keywords: Spacecraft · OBDH software · Automatic code generation

1 Introduction

OBDH subsystem is the information management center of spacecraft. It is responsible for data acquisition, transmission and processing within the spacecraft, between spacecraft and ground station, between spacecrafts. It is also the key to improve the spacecraft efficiency [1]. OBDH software completes the above services with the support of hardware equipment. Therefore, the efficiency and quality of the development of OBDH software greatly affects the development progress and reliability of the spacecraft.

The role of OBDH subsystem in spacecraft determines that it has data transmission interface with all the subsystems in spacecraft that generate data information, as shown in Fig. 1. After the delivery of OBDH subsystem, the spacecraft AIT test will begin. This requires the earliest delivery of OBDH subsystem, which greatly reduces the development cycle of OBDH software. Because of so many interfaces, requirements of OBDH software vary greatly, which leads to the large number and high frequency of changes in the development process.

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 275–282, 2019.
https://doi.org/10.1007/978-3-030-19153-5_28

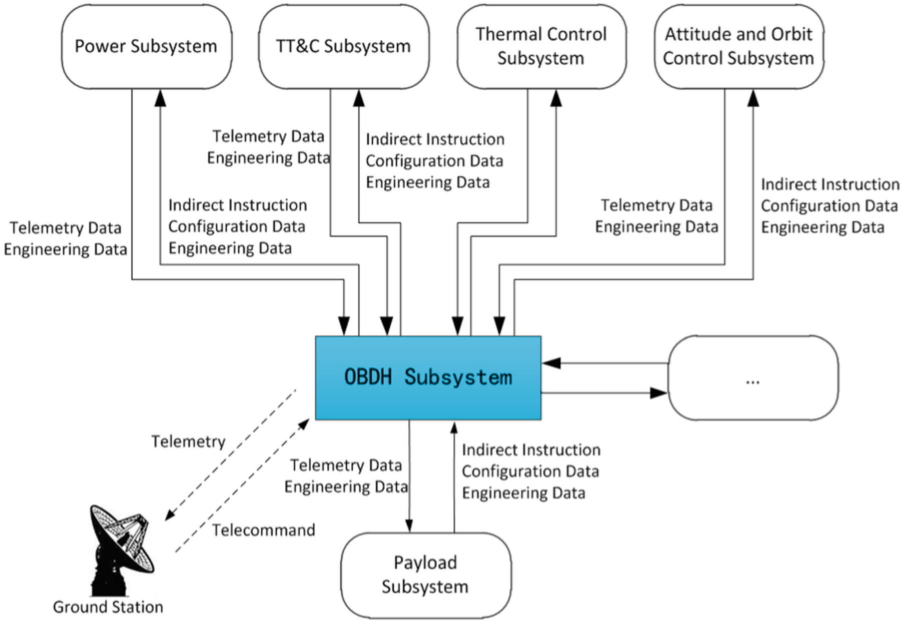

Fig. 1. The external interface diagram of OBDH subsystem

The requirements of OBDH software are mainly Microsoft Word documents, whose content are described by natural language, and the inherent semantic fuzziness of natural language may lead to the ambiguity in requirement description [2]. A large number of spacecraft data input and output interfaces, data formats, communication protocols, etc., are embedded in Word documents in the form of tables. Due to the complex format constraints in Word tables, the process of transforming these contents into software code by manual coding in the past is inefficient and error-prone. A large part of OBDH software function requirements are configuration parameters, and the change of configuration parameters are frequent. Logic and parameters are strongly coupled in common software architecture, this makes the parameter part changes when would inevitably cause logical parts changes, this leads to changes in the risk of unnecessary mistakes.

In this paper, an automatic code generation method based on electronic data sheet (EDS) is proposed. The requirements are provided in the form of EDS. EDS is used to format spacecraft data source, data channel, data format and transmission protocol. Programmers can use special tool to convert the output of EDS into software code, greatly reducing the time of manual coding. Because the spacecraft data source is managed uniformly in EDS, the inevitable error rate during the manual input process is greatly reduced.

2 Design Status of OBDH Software

2.1 Information Processing Function of OBDH Software

OBDH software is responsible for the information processing of the spacecraft, the efficient and reliable management and control of the spacecraft information flow, the monitoring of the status of each subsystem, and the completion of in-orbit tasks and parameters configuration with the payload, so as to achieve the predetermined objectives and meet the mission requirements. OBDH subsystem adopts the bus mechanism to connect each subsystem organically and realize the exchange and sharing of information and data to complete the management, control and task scheduling.

The information processing function of OBDH software includes:

(1) Telemetry function: collecting telemetry data of OBDH and other subsystems, and then process and store them. It is formatted according to the constraint, and transmitted to the ground via the telemetry downlink radio frequency channel.
(2) Remote control function: receiving remote control commands and uplink data from the ground station, decoding, verifying, distributing and executing them, and output remote control commands for other subsystems.
(3) Bus management function: at present, most spacecraft adopt 1553B bus or CAN bus, and OBDH software completes the communication management between the subsystems attached to the data bus, controls the data transmission between the subsystems, and completes the spacecraft mission.

Other functions of OBDH software, such as program control function, housekeep function, autonomous management and autonomous operation function, etc., are not described in this paper.

2.2 Structure of OBDH Software

Most of the requirements for the information processing function of OBDH software come from various protocol tables, which are generally given in the form of Microsoft Word documents. Programmers copy the required protocol information from the Word documents and forms the software code after symbol and format conversion, which is generally expressed as various parameters. In the current software structure, logic and parameters code have high coupling, as show in Fig. 2.

When the parameter part of the code needs to be changed, programmers need to copy the data from the new version of the protocol table, after conversion, the software code is formed. Due to the coupling between the logic code and the parameter code, the logic code may be changed when changing the software parameters, thus causing software exceptions.

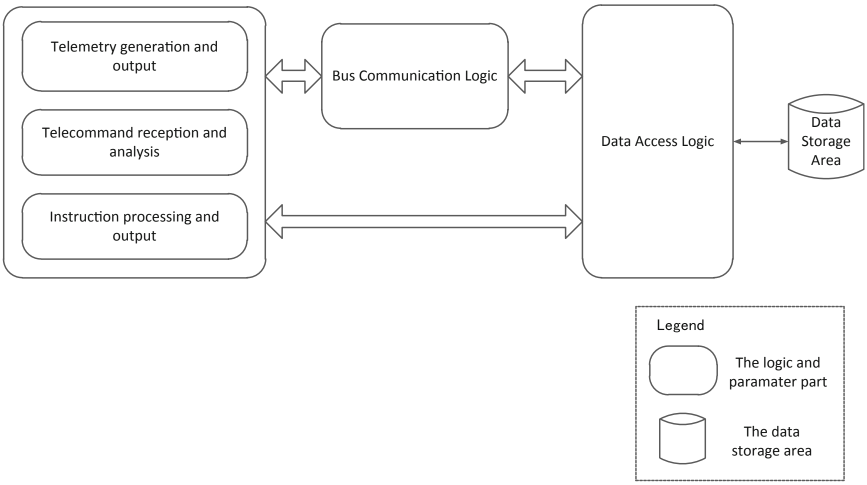

Fig. 2. A software structure coupled with logic and parameters

3 Automatic Generation of Software Code

3.1 Software Structure that Separates Logic from Parameters

Since the requirements of OBDH software have changed a lot and the changes are mostly related to information flow, the software changes caused by these changes are mainly the modifications of configuration parameters, which basically do not involve the changes of software logic. Therefore, a software structure that separates logic from parameters is designed, as shown in Fig. 3. The logic part designs the standard interface, calls various constraints of the parameter part, and completes the acquisition and processing of spacecraft information flow.

The logic part according to the constraint relation of the parameter part, completes the corresponding data acquisition, processing and transmission. For example, the logic part completes the bus communication control according to the bus communication protocol agreed in the parameter part, the acquisition of telemetry data is completed according to the telemetry parameter acquisition channel agreed in the parameter part, the generation of telemetry packages is completed according to the configuration of telemetry packages and the type of parameters agreed in the parameter part, the instructions output is completed according to the instruction format agreed in the parameter part.

The logic part uses mature code, so the spacecraft development process does not need to re-develop the logic part of OBDH software. The parameter part is generated automatically by converting the output of EDS system.

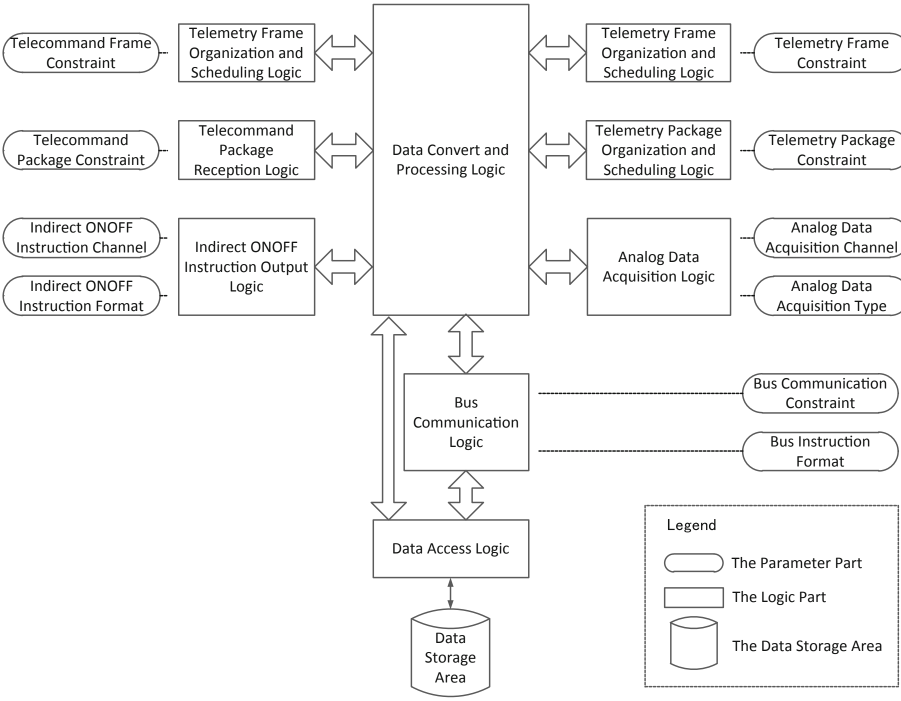

Fig. 3. Software structure that separates logic from parameters

3.2 EDS System

EDS is short for electronic data sheet. It can realize the full-cycle management of the spacecraft electronic information system design. It satisfies the coordinated design and transmission of information overall, OBDH subsystem, single device, AIT, operation control, etc.

EDS integrates the input and output data sources of spacecraft information overall to form a unified basic data source of spacecraft. Data classification and hierarchical management are realized, and data association related to information flow at the spacecraft system level, subsystem level and configuration item level is realized. This ensures the spacecraft in the overall information design, software development, measurement, operation control, simulation and verification and ground testing phase of effective synchronization, as shown in Fig. 4. In this paper, the management parameters provided by EDS are used for OBDH software development, as shown in the dotted line box in the figure.

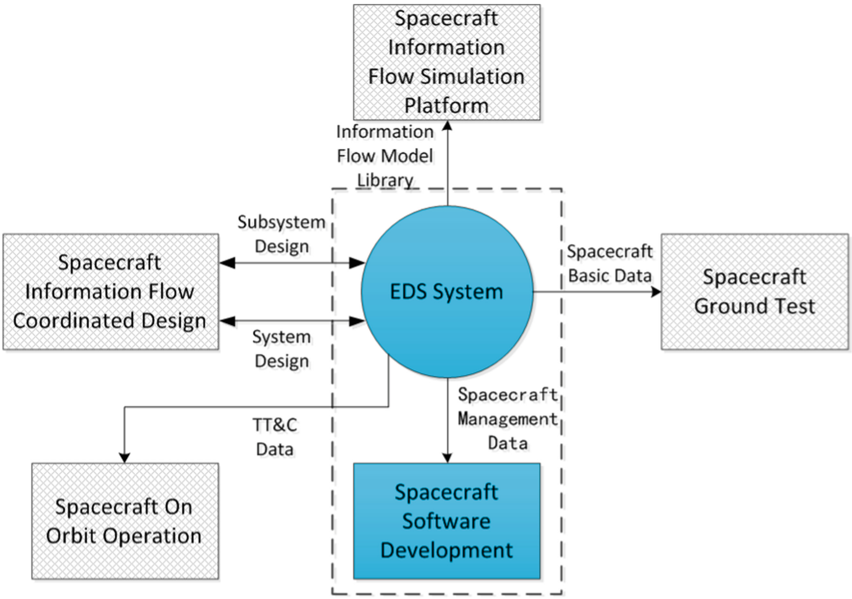

Fig. 4. EDS interface diagram.

The data that EDS can output includes telemetry parameter acquisition channels, remote command channels, telemetry package, telemetry frame, virtual channels, telemetry packet scheduling strategy, telemetry frame scheduling strategy, bus transmission protocol and so on. Here are two typical data constraints to illustrate (Tables 1 and 2).

Table 1. Analog telemetry acquisition constraints

Parameter name	Instructions
Telemetry parameter code	Uniquely identifies telemetry parameter
Telemetry parameter name	Description of telemetry parameter
Telemetry parameter type	Including analog, temperature and BL
Acquisition device ID	Identifies the device that collects remote data
Acquisition channel	Identifies the acquisition channel of telemetry data
Acquisition length	Indicates the length of current telemetry data

Table 2. 1553B bus protocol constraints and examples

Sub address	Word count/mode code	T/R	Communication mode	Transmission cycle(s)	Service request	Data content
7	32	1	RT -> BC	1	0	Telemetry Data1
8	32	1	RT -> BC	1	0	Telemetry Data2
20	32	1	RT -> BC	1	0	GPS original observation data
21	128	1	RT -> RT	0.5	0	Orbital measurement data
24	32	0	BC -> RT	–	1	GPS import data
31	17	0	BC -> RT	–	0	Synchronization word

3.3 EDS Output Conversion

XML (Extensible Markup Language), was founded in 1998 by the main design organization the W3C (the World Wide Web Consortium) of the world wide web. It is a Markup Language, its format to describe the structure and semantic clarified. It is a plain text, non-proprietary format, and you can use any editor to edit, can display in a variety of devices [3]. It is effective to avoid the Word document is limited by a proprietary format.

The EDS system outputs XML files in uniform format, including analog telemetry acquisition type and channel, indirect ONOFF instruction output format and channel, bus instruction format, bus communication protocol, telemetry packet format and content, telemetry packet scheduling strategy, telemetry frame format, telemetry frame scheduling strategy, virtual channel allocation, etc. As it comes from EDS system, the changing process is selected and completed in EDS system. Taking it as the requirement input for the development of OBDH software, it can effectively guarantee the uniqueness of data source and avoid the error caused by inconsistent document versions.

Programmers dedicated conversion tool converts the XML files to the software code, this part of the software code stored in a separate c file, after the follow-up requirement change also change the c file only, do not change the software logic code section, reduces the possibility of software failure caused by accidental modification of the logic code when modifying the parameter code.

Take telemetry package format as an example to illustrate the software code generated by the output of EDS system.

```
struct tm_packet_struct
{
    unsigned int tm_index;
    unsigned char tm_type;           /*Three types: byte type， double level type
and bit type*/
    unsigned int tm_length;          /*Telemetry parameter length in bits*/
    unsigned char bit_pos_in_byte; /*The start position of a bit-type telemetry
parameter in a byte*/
};
```

4 Conclusion

Due to the numerous external interfaces of spacecraft OBDH subsystem and frequent changes in requirements, the manual coding of OBDH software is low efficient and reliable. An automatic code generation method is proposed based on EDS. The software development process is illustrated. This method can more significantly reduce the duplication of work caused by frequent changes in requirements, avoid the software failure caused by manual coding errors, and improve the reliability of OBDH subsystem and even the spacecraft.

References

1. Weichi, T.A.N., Yingqi, G.U.: Space Data System, pp. 8–9. China Scientific and Technology Press, Beijing (2004)
2. Xiong, Q., Yang, H.: Application of detects seeking for evaluating the ambiguity in software requirements documents (SRS). Wirel. Commun. Technol. (2), 38–42
3. Yuan, Y., Liang, S.: A brief introduction to XML Extensible Markup Language. Comput. Knowl. Technol. **6**(20), 5523–5526 (2010)