Structured Design Method of Spacecraft Telecommand Information Flow Based on Electronic Data Sheet

Nan Pei(✉), Shuo Yang, Yanfang Fan, Yong Xu, Qiang Mu, and Hongjing Cheng

Beijing Institute of Spacecraft System Engineering, Beijing 10094, China

Abstract. This paper proposes a structured description method of spacecraft telecommand based on electronic data sheet. The collaborative design is realized by constructing command tree. The command tree model is composed of multi-level data sheet, which contain component combination and interface. A basic component is the smallest unit of command that describes data fields by standard types and attributes. The EDS system and general command generation tool are developed to implement the application of the model. With the characteristics of clear hierarchy, flexibility and high expansibility, it solves the problems of asynchronous information and understanding differences, reduces the repeated generation and check work in telecommand application, and improves the work efficiency and correctness.

Keywords: Electronic data sheet · Structured design · Component

1 Introduction

With the development of space missions, the system functions, information flow, protocols and interfaces have become more and more complex, and the scale of commands has also been increasing. Generally, format requirements and design results of telecommand are described through documentation. There are the following problems. Data is inconsistent in different documents, and error is occurred in unstructured data transmission, and a lot of understanding and input work is repeated. These problems lead to decreased efficiency and increased errors [1].

According to the requirements of standardized and structured information design and transmission, relevant research have been carried out. The SOIS Electronic Data Sheets (SEDS) [2, 3] and XML-based Telemetric and Command Exchange (XTCE) protocol which proposed by Consultative Committee for Space Data System (CCSDS) [4] are more authoritative.

SEDS focus on the definition of data interaction interface for satellite-borne information flow, while the XTCE focuses on the data interaction interface between the satellite and the ground station.

This paper proposes a structured representation of command based on model. In the process of information flow modeling, the specification defined by SEDS and XTCE

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2022
Published by Springer Nature Switzerland AG 2022. All Rights Reserved
Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 186–193, 2022.
https://doi.org/10.1007/978-3-030-93398-2_18

standards is greatly borrowed. The command information is divided into multiple fields and abstracted to form a model. A single-layer command format is formed through the combination of basic models, and then a tree structure is formed through the relationships among levels to achieve data structuring. To complete the structured model, a visual data sheet is provided to designers in a collaborative design tools. Designers of different systems can use data sheets to realize data interaction. Finally, the results can be transformed into standard interactive interfaces to realize structured transmission.

2 The Process of Spacecraft Information Flow Design

The process of information flow design is a V-shaped structure which transmit requirements from top to bottom and collect data from bottom to top. It is also the realization of information interface between various protocol layers according to SOIS [5, 6].

In system-level, the telecommand protocols of data link layer and transport layer are designed and the interface is identified. Requirements for the services and entities are passed through the interface, such as space packet service in transfer layer and PUS in application support layer.

Requirements are passed layer by layer to the final design entity. The design entity may be a device or software. After the completion of the design, the telecommand information from the entity will be transmitted upward, including the format, coding, using conditions and criteria. After the step-by-step transmission, a complete command information flow is formed.

According to the above procedures, a structured and visualization model is provided by electronic data sheet for command protocol design. The design results are eventually converted into XML files which comply with XTCE and SEDS standards. The output files can be read directly by the computer system of user, which can greatly improve work efficiency and effectively standardize the design.

3 Design of Electronic Data Sheet

3.1 Design Principles

The frequently used spacecraft remote control standards including PCM telecommand, the packet telecommand and AOS telecommand. Hierarchical design is used in all of three, and packet is defined and used in packet telecommand and AOS telecommand. Although the content of the command is not defined, the interface between each other is standardized because all users work with packets [7].

The European Cooperation for Space Standardization (ECSS) has developed a standardized telemetry and command format, ECSS-E-70-41A Packet Utilization Standard (PUS) [8], which defines specific types and contents through service types and sub service types.

The electronic data sheet should fully considers the above standard protocols, and follows the principles below:

Flexible. The existing protocol design should be covered.
Extensible. Future extensions of new protocols should be considered.
Structured. The interface data is machine-readable to avoid ambiguity.

3.2 Design of Basic Component

Component is defined as the smallest unit that makes up the command format. The basic element is shown in Fig. 1. The attributes of component include name, identifier, length, and type. The name attribute describe the component meaning, such as block head, command type, etc. The identifier attribute is the universally unique identity, which can be referenced. The length attribute is the length of the Component in bytes. The type attribute is used to distinguish between different basic component. Different types of component have different extended attributes, including algorithm, value, range and extended information.

According to the analysis of the commonly used command structure, the types of components are divided into seven categories: enumeration, constant, check, padding, count, data field and user input.

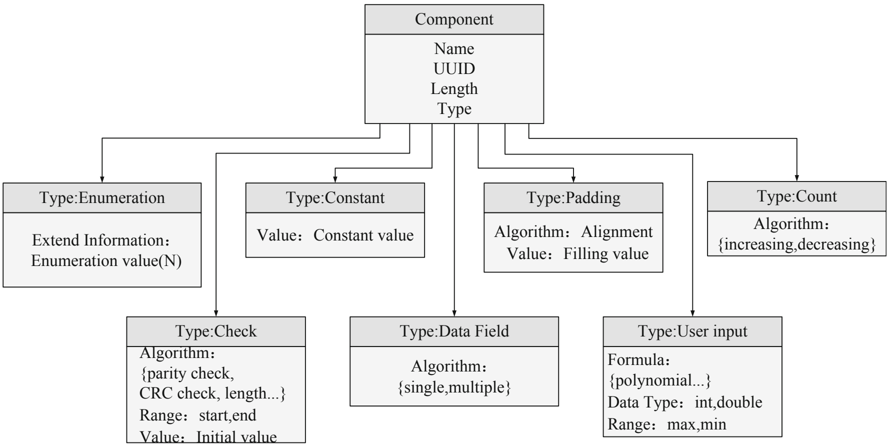

Fig. 1. Types and attributes of basic component

Enumerated types are used to describe units with a finite number of classes which can be extended as interfaces to lower-layer, such as the APID in the packet. The extended attributes of enumeration type include enumeration value and meaning. The enumeration value and meaning correspond one to one, and there can have multiple enumeration descriptions.

Constant type is used to describe a fixed data unit, such as a fixed header in a command. The extended attribute of constant type only contains the value, which is the fixed value of the data unit.

Check types are used to describe a unit computed by some algorithm from other units. Such as checksum or data field length and so on. The extended attribute of check types include algorithms, ranges and values. The algorithm includes parity check, CRC, ISO, length by byte, length by word, etc. It can be expanded according to the need. The range is used to identify the calculated range, such as calculate length from start to end. The value is optional and is used as the initial value of the algorithm.

Padding type is used to describe the filling data unit to reach the required length. The extended attribute include algorithms and values. The algorithm describes the filling method, such as filling by even bytes, filling by fixed length, etc. The field value is the data value of padding.

Count type is used to describe the unit of data that generates the count, such as the sequence number. The extended attributes of the counting type only includes algorithms, such as increasing and decreasing, which can be expanded according to needs. When an command code is generated, the count type fields can be counted and changed as needed.

Data field type is used to describe a unknown data unit, such as the packet data field of an packet. The detail of data field type will be described at the lower-level. The extended properties of the data field type include algorithms, used to select whether multiple data unit can be included in this data field.

User input type is used to describe a data unit which value is input by user instantly when the command is generated. The extended attribute of user input type includes formula, data type and valid range. The specific value will be obtained by real-time calculation when the command is generated according to these attributes.

3.3 Design of Container Data Sheet

An electronic data sheet for a physical device or a software is defined as a container that consists of an ordered series of components.

The container can be composed of name, ID, affiliated device entity, component combination and interface. ID can be uniquely referenced. Affiliated device entity is used to describe the specific position in the system. Components are combined into an ordered set that describes the format of the command. Interfaces include key-value to specify branches in the telecommand tree.

3.4 Design of Interface Connection Between Containers

Considering the attributes and design flow of telecommand information, a tree structure is formed through multiple nodes.Each node corresponds to a container. The root node reflects the top-level information. The intermediate node can have one father node and several children node. The branching relationship between container is defined by the key-value. Leaf nodes describe the command code.

Containers are connected through interfaces shown in Fig. 2. The upper layer container contains a data field type unit as a data association and a set of values for the enumeration type unit as all key values. The number of branches is the same as the number of key values. The child container selects one key value to form a branching relationship. If the parent container does not include enumeration type, only one branch can be connected.

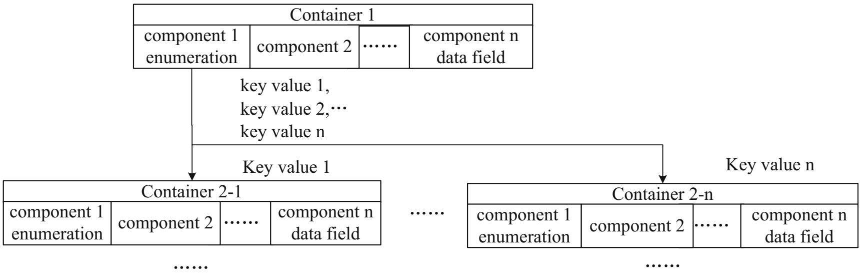

Fig. 2. Branch connection relationships between containers

When there is no data field type component in a container, it means that this is the lowest level. Generally, fixed code are described by enumerating fixed command, and variable parameter are described by using user input component. The formula, data type and other extended information are used for calculation when generating command.

4 Application of Electronic Data Sheet in Telecommand Information Flow Design

4.1 Design and Application Based on Layered Model

The telecommand design process is completed by constructing the command tree. The top-level user establishes the root sheet, design the rules into the container structure, and pass it to the lower-level user. The lower-level user, refines the command design in this level. Due to the restriction of structure, the lower-level users can only design in strict accordance with the requirements of the higher level. Finally, the bottom-level user completes the leaf sheet design to close the branch of the tree. After all branches are closed, the complete command tree is constructed, including all command formats, code, and rules.

The layered tree structure has strong extensibility, and conforms to the design method that SEDS describes the device information and service interface information in SOIS architecture.

The interface is connected through the format of command and key values between device entities, and the data interaction between protocol layer is realized. It is suitable for collaborative design among multiple users and can realize the design requirements of “Plug and Play” for spaceborne equipment. New business models and applications of spacecraft information flow design can be easily and flexibly expanded.

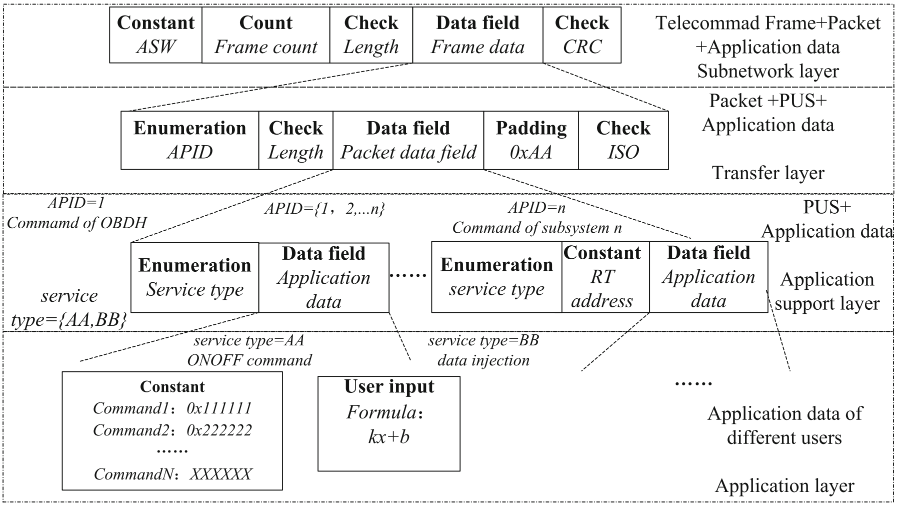

Fig. 3. Example of instruction tree model construction

The application example is shown in the Fig. 3. Different terminals in the application layer have their own application data format. The application data is packaged in application support layer using the PUS protocol. Telecommand packet of different APIDs if formed in network layer and transferred to data link layer. This process represents the interface and relationships of different levels.

Collaborative Information Flow Digital Design Platform which called EDS (Electronic Data Sheet Manage System) has been developed by CAST based on this model. The EDS system provide information collection, formatting, sharing, change management and other functions for spacecraft information flow design and verification [9]. At the same time, it provides third-party data interfaces for applications, including spacecraft software design [10], system test and in-orbit operation.

The formatted data sheet is designed in EDS system. Designers record their design result in the database through distributed sheet filling. Meanwhile, the system provides designers with a unified data source to ensure that the design data are uniform, accurate and clear for all users.

The electronic data sheet provides the interface for users, which is shown in Table 1. Each row is a component, and multiple rows form a container. The user can select the component type through the pull-down menu, and fill in the attribute information. The system can automatically verify the information, to ensure the design is correct. After the completion of the filling, the system will automatically generate the model and pass it through the interface.

Multiple data sheet are connected to form a command tree. Each node in the tree is an electronic data sheet. The system converts the complete command tree to form a standard interface file output.

Table 1. Data sheet of command format.

Component name	Length (Byte)	Type	Algorithm	Range	Value	Enumeration details
Packet version number	0.375	Constant	/	/	0	/
Packet type	0.125	Constant	/	/	1	/
Secondary head flag	0.125	Constant	/	/	0	/
APID	1.375	Enumeration	/	/	/	0x401:real-time 0x402:delay
Sequence flags	0.25	User input	/	/	/	/
Sequence count	1.75	Count	Increasing	/	/	/
Packet data length	2	Check	Length by byte	[8, 8]	/	/
Packet data field	0	Data field	/	/	/	/

4.2 Output and Application of Structured File

In the system design, integration test, the research of telemetry information design and processing based on XTCE has been carried out and applied [11, 12].The electronic data sheet of command format generated in the different layer is gathered into an command tree, which is converted by software tools to generate XML (Extensible Markup Language) interface files for transmission. The structured information is transmitted to software design platform, sub-system test platform, integrated test system, on-orbit data management system and other application scenarios.

The structured output file is in good agreement with the XTCE definition. By defining reference type, set and container, XTCE standard describes all attribute information of container, parameter and argument used in telemetry and telecommand. The interaction interface of spacecraft information flow data is completely defined.

First, standard and custom types are defined. And then, the description of the set defines all the parameters and arguments used by the spacecraft and associated then to the types. Finally, the parameters or arguments are arranged in order to form the command format.

The component model defined in this article corresponds to the type sets including ParameterTypeSet and ArgumentTypeSet in XTCE. The instantiated components that are filled into the data sheet with specific parameters corresponding to the data sets including ParameterSet and ArgumentSet in XTCE. The data sheet corresponds to the Container in XTCE, and the command tree corresponds to the ContainerSet. Finally, command code corresponds to MetaCommand in XTCE.

Standardized command generation tool is developed, which can display the structure of telecommand after read and transfer the XML file. The tool can generate commands automatically and quickly. The command code can be reverse solved according to structured format using to be checked. In this way, a general tool can adapt to multiple risks without repeated development and configuration. It will play an important role in testing and on-orbit operating.

5 Conclusion

In this paper, a telecommand information model is proposed based on layered assembly model. Different types of basic components are combined into data sheet and connected through interfaces to form a tree structure. The electronic data sheet is filled through multi-level collaborative design. With the characteristics of clear hierarchy, flexibility and strong expansibility, it solves the problems of data asynchronization and understanding of differences, reduces the repeated generation and check work, and improves the work efficiency and correctness.

Later, different levels of interface parsing methods can be developed to achieve plug-and-play and automatic code generation.

References

1. Jonathan, W.: Using CCSDS Standards to Reduce Mission Costs. NASA Goddard Space Flight Center, Greenbelt (2017)
2. CCSDS. 876.0-R-3 Spacecraft onboard interface services - XML specification for electronic data sheets: Washington, CCSDS (2018)
3. CCSDS. 660.0-B-1 XML telemetric and command exchange (XTCE), Washington: CCSDS (2017)
4. CCSDS. 876.1-R-2 spacecraft onboard interface services—specification for dictionary of terms for electronic data sheets, Washington: CCSDS (2016)
5. Yang, L., Chen, B., Zhang, R.: Application of SEDS in service and protocol system. Comput. Measur. Control **26**(11), 248–251 (2018)
6. Zhang, X., Lyu, L., An, J.: Instance design of EDS based on SOIS. Comput. Eng. Des. **41** (9), 2670–2677 (2020)
7. He, X.: Design and application of a common spacecraft telecommand data format. Spacecraft Eng. **17**(1), 94–99 (2018)
8. European Cooperation for Space Standardization: ECSS-E-70-41-A Space engineering: ground systems and operations telemetry and telecommand packet utilization. Noordwijk, ECSS (2003)
9. Guo, J., Li, R., Fan, Y., et al.: Network coordinated digital design and practice of spacecraft information flow. Spacecraft Eng. **29**(4), 59–65 (2020)
10. Zhang, H., Pan, L., Yu, M.: A method of automatic code generation for spacecraft OBDH software. In: Jia, M., Guo, Q., Meng, W. (eds.) WiSATS 2019. LNICSSITE, vol. 280, pp. 275–282. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19153-5_28
11. Liu, Y., Li, Z., Ding, X., et al.: Satellite telemetry data processing method based on XTCE. J. Telemetry Tracking Command **38**(2), 27–31 (2017)
12. Zhang, H., Guo, J., Kuang, D., et al.: Design of spacecraft telemetry transfer universal interface. Spacecraft Eng. **28**(6), 46–51 (2019)