Spacecraft Electronic Datasheet Based Onboard Spacecraft Test System Design

Yanfang Fan(✉), Hongjing Cheng, Ruijun Li, and Zhang Tao

Institute of Spacecraft System Engineering, Beijing, China

Abstract. To reduce spacecraft development costs and risks, improve quality of building and test equipment, increase spacecraft software reuse and test system reuse, the paper presents how to develop a SEDS based test system that can automatically extract SOIF and protocol information from SEDS to build an efficient and organized testing environment. The test system established by this method is consistent with the concept of model-driven software organization. Practice has proved that this test system architecture can effectively use the information transmitted by SEDS to establish a spacecraft difference free test environment, which can not only save labor cost, but also boosts manufacture efficiency, and improves test quality by improve the quality of test basis input.

Keywords: Spacecraft · Spacecraft electronic datasheet · Test system · Model

1 Introduction

In the past, each satellite has its own principles to design hardware/software interface, so that, during the spacecraft design and implementation process, the interface function and compatibility between different devices and/or subsystems cannot be verified until most of integration work complete, which results in identifying potential problems. However, nowadays, the payload and platform equipped by spacecraft is becoming more and more complex, and the manufacture period requirements are getting shorter and shorter, the satellite manufacturing industry must transform from the classical type to productization.

Spacecraft manufactures tried multiple methods to achieve productization. One of the most effective tries is known as Spacecraft Onboard Interface (SOIF) [1]. SOIF is a serial of standards for the interchange of information, and the interconnection of subsystems and devices onboard of a spacecraft. The standardization will allow for the enhanced reuse of spacecraft equipment and software, which leads to the productization [2]. CCSDS has been branching out to provide new standards on SOIF for many years [3]. Recently, Chinese satellite manufacturers began to engage in this research too, and has made a lot of meaningful attempts [4]. For example, in order to mandate manufactures following and apply the SOIF, Chinese Academy of Space Technology (CAST) has developed an information system called spacecraft electronic datasheet system (SEDS) that documented the SOIF and standard protocol specifications in terms of an electronic datasheet, and provide XML file for the design data exchange and utilization.

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 201–206, 2022.
https://doi.org/10.1007/978-3-030-93398-2_20

To maximize the productization advantage, our team developed a modular layered test system, which will firstly extract structure information from the Spacecraft electronic datasheets and manual freely create appropriate processing components, and secondly connect each component together according to the SOIF. The whole testing environment creating procedure is not only a procedure of instantiation of spacecraft datasheet model, but also a digital simulation process, which will provide verification environment for the correctness of the information exchange protocol and simulate the compatibility of each real device interface.

The paper proposed a SEDS based test system, which design processing components' interface according the stipulation of the SOIF. Simultaneously, it can simulate and verify the consistency between the realization of the onboard hardware/software and with their user requirements. The following sections will explain how the component models of the test system have been created through the SEDS, and how the test system will be established through these components. The evaluation of the trial effect of the system will be given at the end of the paper.

2 SEDS Based Modeling

In order to effectively model the information provided by the SEDS, we observe the SEDS through three orthogonal views. The first view is the Protocol view, which describes the protocols and services that are to be implemented in the onboard spacecraft system in order to provide the users with the advantages of the SOIF architecture. The second view is the Services View, which describes the data communications services that are provided to the users. And finally, the Interoperability view, which describes how the data has been exchanged between different spacecraft data busses.

From the perspective of Protocol view, the process of Onboard software development can be described as a nested process. On the contrary, the process of data processing by the test system can be modeled as a de-nesting process. The correspondence between hierarchical transmission protocol of the spacecraft and data process component model shows in Fig. 1.

The Protocol view model abstracted information flow as a list of data package, which was composed with data segments and sub-data packages. Each layer can be modeled as a data process component. The function of this component is to decompose the current package to data segments and sub-packages, and assign corresponding attributes to them.

From the perspective of Service view, each application programming interface must bond to a set of procedure and function calls to access services offered by SOIF. From this view, the underlying hierarchy is not only not visible, but not of interest. Users see only a set of APIs that are uniformly accessible from each application. Hence the test system framework needs to automatically create a runtime testing instance which is composed by a set of service components or data processing components, according to the information provided by the SEDS. Therefore, the Service view model abstracted onboard services as a set of components which correspond to the service access points exposed by the SOIF stack. This services view of SEDS Model is shown in Fig. 2.

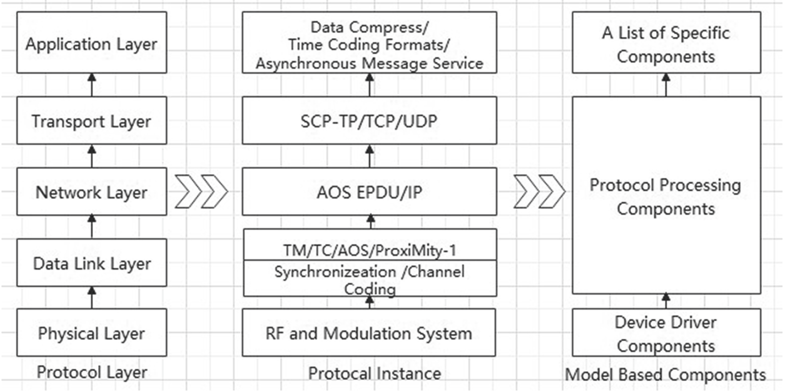

Fig. 1. The correspondence between protocol and component model

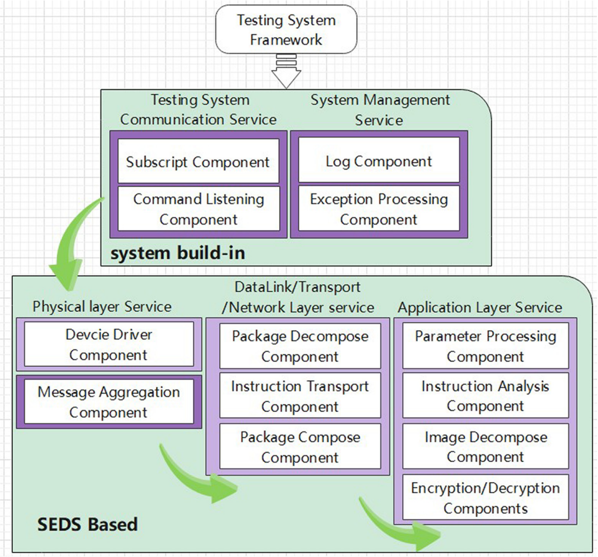

Fig. 2. The services view of SEDS model

The Interoperability view is different from the other two views, it doesn't consider what the test system can do, but how to design a SOIF compliant hardware-driver components, which will be instanced to connect any real hardware device or virtual hardware to a runtime instance. Therefore, the Interoperability view model abstract spacecraft hardware interfaces as a couple of standard hardware-driver components, and the test system framework knows how to extract information provided by the SEDS to instance an appropriate hardware-driver component to meet the needs of a particular testing request. The Interoperability view of SEDS Model is shown in Fig. 3.

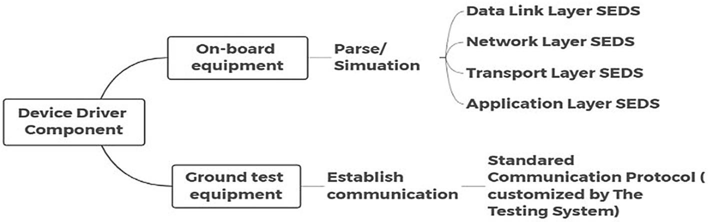

Fig. 3. The Interoperability view of SEDS model

3 System Architecture

The test system is composed of the system framework and a set of components that with different interfaces and functions.

The System framework is the core part of the test system, by which the whole system is controlled, managed, and monitored. It is mainly used to complete the tasks which including loading SEDS, managing instanced components, rendering UI, handling system exceptions and logging system logs, storing test data, etc. All the tasks that the framework is responsible for are system-level and have nothing to do with the testing business work.

Each component was encapsulated as a dynamic library, which possess different interfaces and functions. As described in Sect. 2, there are twelve different categories of components. Firstly, the framework loads the physical layer SEDS that specifies what kinds of device components will be instanced. The interface would correspond to the SOIF of the target machine which is under testing. Besides, the interfaces of each component are depicted by a couple of properties called interface descriptor, including interface type, name, and especially, the successor SEDS unique identify number. This SEDS unique identify number will be used to locate the successor component of the current device component. Then the system framework will instance the next component whose input interface will perfectly match the previous component’s output interface. Similar processes will continue, until all the SEDS have been loaded sequentially, and all the components have been instanced correspondingly. Finally, according to each components' interface descriptor, the framework connects all the components together. In other words, as long as the components are instanced based on the SEDS, the test system would run exactly as the reverse sequence of the onboard system runs. The process of establishing a test instance is shown in the Fig. 4.

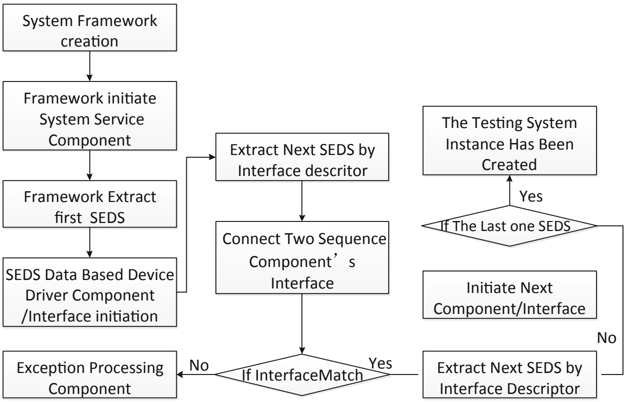

Fig. 4. Schematic diagram of establishing a test instance

4 Experiment and Effects

The SEDS based test system has some distinct properties as follows:

1. No operator is needed in front of the test system establishment.
2. Subjectivity aroused by human data reading is avoided.
3. Realize a high efficiency, one-key automatic test system establishment.

Typical application tests are carried out in integrated electronic subsystems functional tests of more than 10 spacecrafts of five different fields. For application layer, currently SEDS includes five types of parameters, which are analog signal collection, Bi-lever electronic signal collection, temperature variable collection, digital signal collection, etc. Each parameter has properties in SEDS, such as name, Code identity, data type, unit, length, byte order, processing function, calibration, etc. For transport Layer, currently SEDS includes 1553B bus transmission protocol. For Network layer, currently SEDS includes AOS EPDU protocol. For Data link layer, currently SEDS includes telemetry, telecontrol, AOS, BCH coding, etc. With the guidance of SEDS, the test system can correctly instance all the service component in order. Each component’s interfaces can correctly reflect design requests of the target machine’s interfaces. The results of experiments show that the test instance composed by these component instances can satisfy the functional test requests, and can provide all kinds of test data products that, from all levels, prove the system implementation correctness.

In addition, as described above, the functions and interfaces of each component are designed in accordance with the common requirements of spacecraft development and the standard interface specifications. The test instance established based on SEDS will work correctly with no interface mismatch faults, as long as there are not any design faults in the SEDS. However, exception warnings are often thrown out during the instantiation process, which happens because the instanced components with interface design faults cannot be connected as expected. Through the way, the test system is helpful in early verification of the spacecraft system interface design.

5 Summary

By performing of SEDS, not only the spacecraft design procedure, but also its testing procedure benefits from the SOIF.

Firstly, this capability of identifying and selecting the best available data source(s) for an application will ultimately enable better design support. With the continuous progress of the spacecraft development industry, the completeness of the spacecraft development standards will make great progress too. Such, the test system components design will continue to improve accordingly. The benefits of early verification feature of the test system will be more obvious.

Secondly, the test system can automatically create a test instance just according to the information provided by the SEDS, without and manual intervention.

Finally, by extracting appropriate interface or protocol data from SEDS, the test system framework can instance matchable device components that can effectively exchange test data between the target machine and the test system, without and additional hardware interface adaption or adjustment.

All these features mentioned above will reduce the costs associated with integration and test, and potentially all the testing case are reusable in further by a similar spacecraft.

References

1. Smith, J.F.: The spacecraft onboard interface standardization activity. In: IEEE Space OPS 2003 Conference, pp.1–7 (2003)
2. Hansen, L.J., Lanza, D., Pasko, S.: Developing an ontology for standardizing space systems data exchange. In: Big Sky Aerospace Conference, IEEE and AIAA, Big Sky, MT (2012)
3. Drenkow, G.: Future test system architectures. In: IEEE A&E Systems Magazine, pp. 27–32 (2005)
4. Jian, G., Ruijun, L., Yanfang, F., Xiangdong, S.: Digital collaborative design method for spacecraft information flow. Spacecr. Eng. **2**, 59–65 (2020)