



Avionics System Architectures for Software-Defined Spacecraft

Xiongwen He^{1,2}, Jionghui Li²(✉), Fan Bai², Xiaoyu Jia², Xiaofeng Huang²,
and Mingwei Xu¹

¹ Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China

² Beijing Institute of Spacecraft System Engineering, Beijing 100094, China
lijionghui@126.com

Abstract. Developing upon open system architecture, Software-Defined Spacecraft, as a new generation of spacecraft, can support payload plug-and-play, application software loading as needed, and system function reconfiguration on demand. Therefore, the Software-Defined Spacecraft is able to solve the design and application limitations of traditional specific-designed spacecraft, making it become a significant developing trend of future spacecraft. Regarding to the software defined spacecraft bus, this paper analyzes the requirements for avionics system on the aspects of space field development, technology development and corresponding applications. Then, combining with the development status, the avionics system architecture of software defined spacecraft is proposed, along with the related key technologies.

Keywords: Software-Defined Spacecraft · avionics system · intellectualization · networking

1 Introduction

The concept of Software-Defined is originated from Software-Defined Radio (SDR). With the rapid development of modern computing technology, the advantages of Software-Defined are becoming more and more prominent. The scope of Software-Defined is also developing and expanding, gradually becoming an innovative architecture, an innovative design concept and an innovative design method, which has penetrated into many fields, even developing to the trend of software-defined everything. The core idea of Software-Defined is to decouple software and hardware through a unified interface. Under the premise of sufficient hardware performance, software developers are responsible for meeting different needs of different users; And hardware developers can focus on improving hardware performance under a unified standard architecture. This way of “software defined” can minimize the development cycle and cost, and can give the flexibility of system function update and upgrade.

There is also a trend of Software-Defined in the space engineering. The vision is that through the Software-Defined design, the hardware bus of spacecraft can be defined as models, in order to improve the adaptability of spacecraft bus to different tasks. Then, software is used to support on-orbit reconstruction and performance upgrading. By Software-Defined method, the traditional spacecraft is transformed into a multi-purpose and network-collaborated spacecraft, which can shorten the spacecraft development cycle, and reduce the spacecraft development cost. So as to enter a new stage of systematic development and global service of spacecraft.

At present, the spacecraft architecture is mostly customized for specific mission. The spacecraft hardware and software design is closely related to the spacecraft application, with hardware as the main and software as the auxiliary. This traditional spacecraft design method has shown the following five shortcomings: (1) The hardware bus of designed spacecraft are different, and different types of spacecraft do not adapt to each other in hardware, and parts of components cannot be interchanged; (2) The software is incompatible with each other, and the application software developed for one model cannot run directly on the spacecraft of another model; (3) The spacecraft cannot achieve on-orbit performance improvement during its life cycle; (4) The development cycle is long, the cost is high, the functions are inflexible, the on-orbit reconfiguration ability is relatively weak, and it is difficult to mass produce; (5) The spacecraft software is non-accessible for third parties due to its closed architecture. In view of the above shortcomings, it is a good way to develop an open system architecture and a novel generation of Software-Defined Spacecraft that supports payload plug-and-play, application software loading on demand, and system function reconfiguration on demand. The concept of Software-Defined Spacecraft mainly includes software-defined bus and software-defined payload [1]. The avionics system is the kernel to achieve spacecraft bus software-defined. Regarding to the software-defined spacecraft bus, this paper first analyzes the requirements of avionics system for software-defined of spacecraft from the dimensions of field development, technology development and application. Then, combining the state-of-art development status, the ideas of avionics system architecture for Software-Defined Spacecraft is proposed with list of supportive key technologies.

2 Related State-of-art

2.1 Software Defined Spacecraft

At present, technologies related to Software-Defined Spacecraft are actively developing. The Eutelsat Quantum series satellites jointly developed by ESA, Eurosat and Airbus Defense and Space are experimental communication satellites carrying software reconfigurable loads. They can adjust the beam coverage, frequency band and power on-orbit, and can also change the orbit position to achieve on-orbit functional reconstruction [2]. SmartSat is a micro/nano software-defined satellite proposed by Lockheed Martin uses virtual machines and multi-core processing to form an elastic architecture, which supports the

rapid assembly of various satellite platforms and enhances the rapid data processing capability. The frequency bandwidth and beam coverage can be adjusted to support mission on-orbit modification [3]. Galactic Sky, a small solid rocket company in the United States, has designed a software-defined satellite named Gsky-1, which uses Galactic SkyTM, a proprietary software-defined satellite technology developed by Galactic Sky, which adds a cloud based intelligent layer to support the scheduling, deploying, monitoring and managing resources and other missions. With the support of CNES and the ESA, Thales Alenia Space Company is developing a new type of “real-time on-orbit reconstruction” satellite, called “Space Inspire”, which aims to achieve real-time mission adjustment and reconfigurations on-orbit, supporting multiple services from video broadcasting to broadband connection, which maximize the efficiency and effective use of satellite hardware resources.

Research institutions in China are also carrying out research on software-defined spacecraft. The conceptual research of Software Satellite based on SDR has started from 2002, which takes the payload based on SDR and defines the payload function through software [4]. An experimental satellite for software-defined technology, called TianZhi-1 was developed by the Institute of Software Chinese Academy of Sciences, and was launched in 2018. It has successfully carried out more than 10 on-orbit experiments by means of software uploading, including intelligent measurement and operation control, intelligent data compression, intelligent information processing and other aspects [5]. In 2019, the TianXiang satellites (China Netcom 1A and B satellites) were successfully launched. The satellite was equipped with a space-based router to perform software-defined network (SDN). Using the features of SDN, the controllers of the micro satellites which are responsible for highly complex routing calculation can be placed on the ground. However, all the mentioned software-defined spacecraft are mainly small satellites, and there is still a lack of systematic design methods and concepts of software defined spacecraft which suit for all fields from the perspective of system, technology and architecture.

2.2 Avionics System of Spacecraft

It can be seen from the relevant development of software-defined spacecraft that the core idea of software-defined spacecraft is to remove the coupling between the software and hardware of spacecraft products, so that spacecraft software can independently evolve, load on demand, and dynamically reconstruct. By this way, more functions can be realized and more missions can be fulfilled without large modification the hardware. To implement decoupled software and hardware, spacecraft avionics systems are meeting the development requirements of the next generation of spacecraft which is to transform from discrete units into integrated multi-function computing platform [6].

The concept of spacecraft avionics systems is introduced from aviation. The development of aviation avionics systems has gone through four generations, which are separated, combined, integrated and advanced integrated [7]. In 2004, Honeywell International borrowed from the architecture of integrated modular

avionics system (IMA) and proposed an avionics system architecture for spacecraft to implement software-hardware decoupling, aiming to provide dynamic reconfigurability and scalability for spacecraft. This architecture divides a physically high throughput computer into multiple virtual machines, and each virtual machine is used as the resource of corresponding software applications, so as to realize the time division of processing resources, the space division of storage space, and the division of input/output interfaces, and improve the effective fault tolerance performance of the system. At the same time, in order to minimize the impact of changes in user applications on the system, both software and hardware adopt hierarchical design [6], so that the architecture supports open application development.

NASA successfully launched the Orion manned spacecraft in 2014 to implement the avionics system using IMA architecture. The Orion avionics system structure selects Time Triggered Ethernet (TTE) as the backbone network [8]. Each device of the system is connected to the network switch, including 18 time triggered Ethernet switches and 46 terminal nodes. The on-board computer (VMC) is used as the processing core of the spacecraft to complete the main functions of the whole device control, human-computer interaction, and system communication. In 2019, NASA proposed the avionics system architecture of the ‘‘Artemis’’ manned lunar exploration program. This architecture is based on the TTE network, which can provide three different types of data services in the same network, which are best effort, rate limited, and time triggered. Europe launched researches on its new generation of avionics architecture, SAVOIR, in 2010. The latest architecture is shown in Fig. 1. At the protocol level, the architecture integrates the standards of the Consultative Committee on Space Data Systems (CCSDS) and European Cooperation for Space Standardization (ECSS). The operating system supports time-sharing and partition,

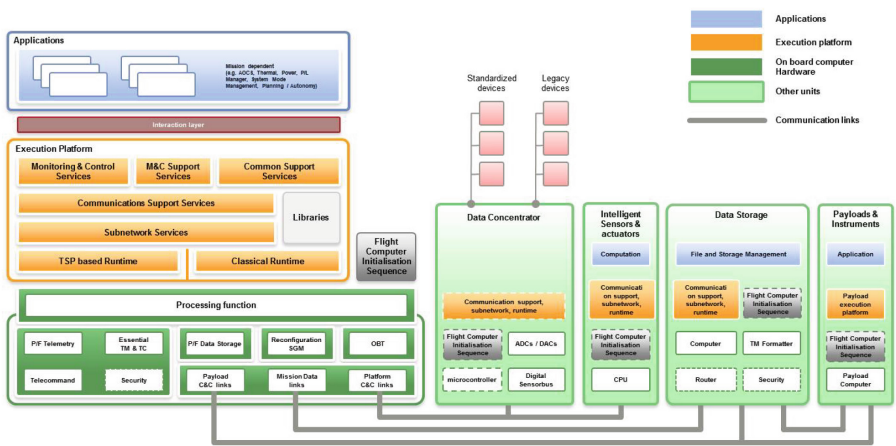


Fig. 1. The avionics architecture of SAVOIR

and the software development mode is to use software components to assemble and realize message communication between multiple application processes through software bus.

In 2015, Zhao Heping proposed that the spacecraft avionics system, as the center of spacecraft intelligence, should have the communication capability of unified information network services, support distributed parallel computing, modular expansion and upgrading, and the computing and storage capabilities of system reconfiguration [9].

It can be seen that the software-hardware decoupling, hierarchical design, fault tolerance and system reconfiguration design, hardware modularization, software reconstruction, time-sharing and partition operating system are the consensus of the next generation avionics systems to supporting the development trend of software-defined spacecraft.

3 Avionics System for Software-Defined Spacecraft

3.1 Features of Software-Defined

The core idea of “Software-Defined” is to decouple software and hardware through a unified interface. Under the premise of sufficient hardware performance, software developers are responsible for meeting different needs of different users. And hardware developers can focus on improving hardware performance under a unified standard architecture. This software-defined method can minimize the development cycle and cost, and give the flexibility of system function update and upgrade.

The core connotation of Software-Defined technology is summarized as follows:

- (1) Software-Defined is a new architecture. The core idea is to implement the decoupling of software and hardware through an interface. The development of software on the interface is not restricted by the hardware. At the same time, the performance improvement of hardware under the interface is not limited to specific hardware functions. The same hardware can be reconstructed through dynamic software programming to complete different functions.
- (2) Software-Defined is a new design concept. In the traditional design concept, the system is designed with the integration of software and hardware, and the software and hardware achieve the customization of system functions with the cooperation of solidification. The design concept of “Software-Defined” is to realize the software of functions on the standard hardware, thus turning the core design of system functions from hardware design to software design. This leap in design concept breaks the situation of customization and solidification of system functions, and realizes the reconfiguration and upgrading of functions.

- (3) Software-Defined is a new design method. Regarding to the hardware design, we can pursue a simplified structure, adopt an open and scalable general platform design, “decouple” from the system functions, and pay more attention to the performance of the hardware platform itself. Regarding to the software design, the component-based design idea is adopted, and the design is carried out according to the requirements of modularization, reconfiguration and upgradability, so as to configure and reconstruct as many functions as possible through software [10].

To summarize the concept, connotation and application of “Software-Defined” technology, we can summarize the development requirements of “Software-Defined” technology as follows: Hierarchical design, architecture standardization, hardware generalization, software-implemented functions and resources virtualization.

- (1) Hierarchical design: Hierarchical design of software and hardware is the premise to achieve functional decoupling of software and hardware. Through an interface (such as an operating system), software definition technology separates software and hardware relatively, breaks the traditional structure and design of software and hardware integration, and realizes the independent development of both. Software can be customized, while hardware is more standardized.
- (2) Architecture standardization: Generally, the operating system is used as the interface, and standardized or open application programming interfaces (APIs) are open to the upper layers and the hardware platform is used through the hardware abstraction layer to the lower layer. At present, several mainstream software-defined architecture specifications have been formed, including the Software Communication Architecture (SCA) specification, the Space Telecommunication Radio System (STRS) architecture specification developed by NASA, and the OpenDaylight software definition network architecture specification.
- (3) Hardware generalization: the hardware here refers to the hardware of the computing and processing platform, not the hardware of physical resources such as antennas. The generalization of hardware lays the foundation for realizing the “decoupling” between system functions and hardware. Therefore, the hardware design should focus on the underlying general computing and processing as much as possible, and try to unbind with the specific functions of the system. Hardware designers focus on the improvement of hardware performance to improve the real-time processing capability of software.
- (4) Software-implemented functions: Software-Defined technology changes the system functions from traditional software and hardware customization to software implementation, which is the most characteristic of software definition technology. Functional software consists of software “components”, which are configured or reconstructed according to different functional requirements and application scenarios.

- (5) Resource virtualization: Virtualization is to provide an abstraction layer between software and hardware by using various technologies, so as to convert physical resources into logical or virtual resources, and enable users, applications or management software running on the abstraction layer to manage and use these resources without having to master the physical details of underlying resources. Therefore, resource virtualization is also a major feature of software definition technology.

3.2 Requirements of Software-Definition Spacecraft

In various spacecraft application fields, the corresponding application requirements put forward some specific requirements for the development of software-defined spacecraft:

- (1) Communication Satellites: the new generation of communication satellites will enter a new era of massive, high-speed, flexible and diversified integration. Under such a development trend, satellites are required to provide a unified on-board software radio universal platform to achieve flexible switching of communication, electronic reconnaissance and electronic countermeasure functions through uploading software or reconstruction. In particular, in the construction of the space-ground integrated network, the communication satellite platform will play an important role as a cloud computing node, provide a unified space based computing and storage platform, and define the network based on software to achieve flexible loading and dynamic access of on orbit functions and provide services for other satellites.
- (2) Navigation satellites: At present, the development trend of various satellite navigation systems in the world is characterized by continuous upgrading of the system, continuous enhancement of autonomous operation capability, multi-task integration, parallel development of multiple enhancement systems, and spiral development of new technologies. BeiDou-3 Navigation Constellation is the largest space-based network on-orbit in China at present. Its network routing update strategy still relies on ground injection, and ground operators are heavily burdened. The subsequent highly efficient and intelligent navigation constellation network requires spacecraft to provide a unified space-based software definition switch platform, support software definition network, and realize network topology construction and network routing customization on demand.
- (3) Remote sensing satellites: With the continuous development of remote sensing payload technology, the mission of remote sensing satellites has begun to change from data acquisition to information acquisition, and has shown a trend of networking and collaborative work. Therefore, subsequent development requires spacecraft technology to provide a unified and universal hardware platform for on-board real-time information processing, and support on-demand loading and upgrading of data compression, target detection, target recognition, cloud judgment and other functions.

- (4) Deep space probes: Long mission cycle and uncertainty of mission environment are the prominent features of deep space missions. For future deep space missions, spacecraft are required to provide a unified on-board avionics hardware platform to support on-orbit loading of intelligent autonomous management related APPs, and to support on-orbit upgrading of autonomous command management, autonomous health management, autonomous communication management as well as autonomous payloads management [11–13].
- (5) Manned spacecraft: With the subsequent construction of space stations and the development of manned lunar exploration projects, from a single manned spacecraft, cargo spacecraft to multi-purpose spacecraft, and even to a multi-purpose platform, and gradually move towards commercial operation, the requirements for the intelligence, flexible scalability, reasonable cost and sustainable development of manned spacecraft are becoming higher and higher, requiring spacecraft to provide a universal on-board computer platform, in order to support flexible migration and reconstruction of multiple functions.

These requirements from the above application fields all focus on the avionics systems of software-defined spacecraft. In addition to the telemetry, remote control, housekeeping, time management, thermal control management, energy management and other functions of the traditional avionics system, the requirements focus on supporting mass production, function definition, spatial networking and autonomous intelligence. The specific requirements are summarized as follows:

- (1) Mass production: In order to adapt to the increasingly fierce market competition, mass production can effectively reduce the design and production cycle of spacecraft, thus reducing costs and increasing profits. For spacecraft, the separation of software and hardware is an effective technical approach to achieve batch quantization. The standardization and modularization of avionics system hardware platform can effectively alleviate the problems of high cost, long development cycle and unstable quality. Compared with hardware, software is inherently characterized by short development, debugging, test cycle and low cost. It is easy to reuse and can quickly meet user requirements.
- (2) Function definition: under the premise of ensuring the safety of spacecraft, the internal resources of avionics systems are fully open and accessible, and the potential of spacecraft hardware can be utilized through software reconstruction. Through virtualization technology, the software modules can be reused to the greatest extent, the development process can be shortened, so that functions can not only be defined, but also be rapidly reconstructed, and users can pay attention to the expanded function APP as needed, so as to adapt to the flexible needs of users, achieve one satellite multi-purpose, and maximize the application of spacecraft.
- (3) Space networking: The avionics systems provide the core components, protocols and software for the integration and interconnection of space and

spacecraft, realizes the flexible networking of spacecraft, the interconnection and interworking of satellites in communication, navigation, tele-control and other fields, and forms the space-ground integrated network (SGIN), which supports the improvement of application efficiency through information fusion and collaboration. With “software-defined” development, the avionics system can quickly complete system resource allocation and response by software according to user requirements (delay, bandwidth, packet loss rate, etc.), and realize the agile reconfiguration of the network (topology change, function deployment, etc.).

- (4) Autonomous intelligence: the avionics system provides mission level autonomous capability and constellation level autonomous capability. Among them, the mission level autonomous capability needs to achieve health monitoring and fault reconstruction, inter-device and intra-device data management, on-orbit resource scheduling, multi-source information fusion through high-performance task management units to reduce the dependence on the ground. Constellation level autonomy is mainly aimed at communication satellites, navigation satellites and remote sensing satellites for formation detection. Through intelligent routing, intelligent forwarding and processing, it can reduce the ground operation and maintenance pressure brought by the increase in the number of spacecraft.

3.3 Avionics Architectures for Software-Defined Spacecraft

Regarding to the requirements mentioned above, the avionics architectures for Software-Defined Spacecraft consists of three parts, which are protocol architecture, hardware architecture and software architecture.

The protocol architecture of avionics architectures for Software-Defined Spacecraft as shown in Fig. 2.

The protocol architecture consists of application layer, transfer layer and subnet layer. Standard protocols of the CCSDS [13] and the ECSS are adopted. With this architecture, different spacecraft can be interconnected, and different equipment inside the spacecraft can also be interconnected, providing a basis for software-defined, task migration on demand and APP dynamic loading.

The main features of the proposed protocol architecture include:

- (1) Space and internal network integration design: The space and internal network integration of low-speed links (measurement and control links and low-speed buses) is realized through the space package protocol of the transfer layer, and the space and internal network integration of high-speed links (data transmission/inter satellite links and high-speed buses) is implemented through the IP protocol of the transfer layer.
- (2) Layered protocols: Protocols at different levels can be flexibly extended, replaced and upgraded.
- (3) Separation of the common-used services and the customized services in the application layer: A large number of common-used services are provided through the application support layer, while functions in the application

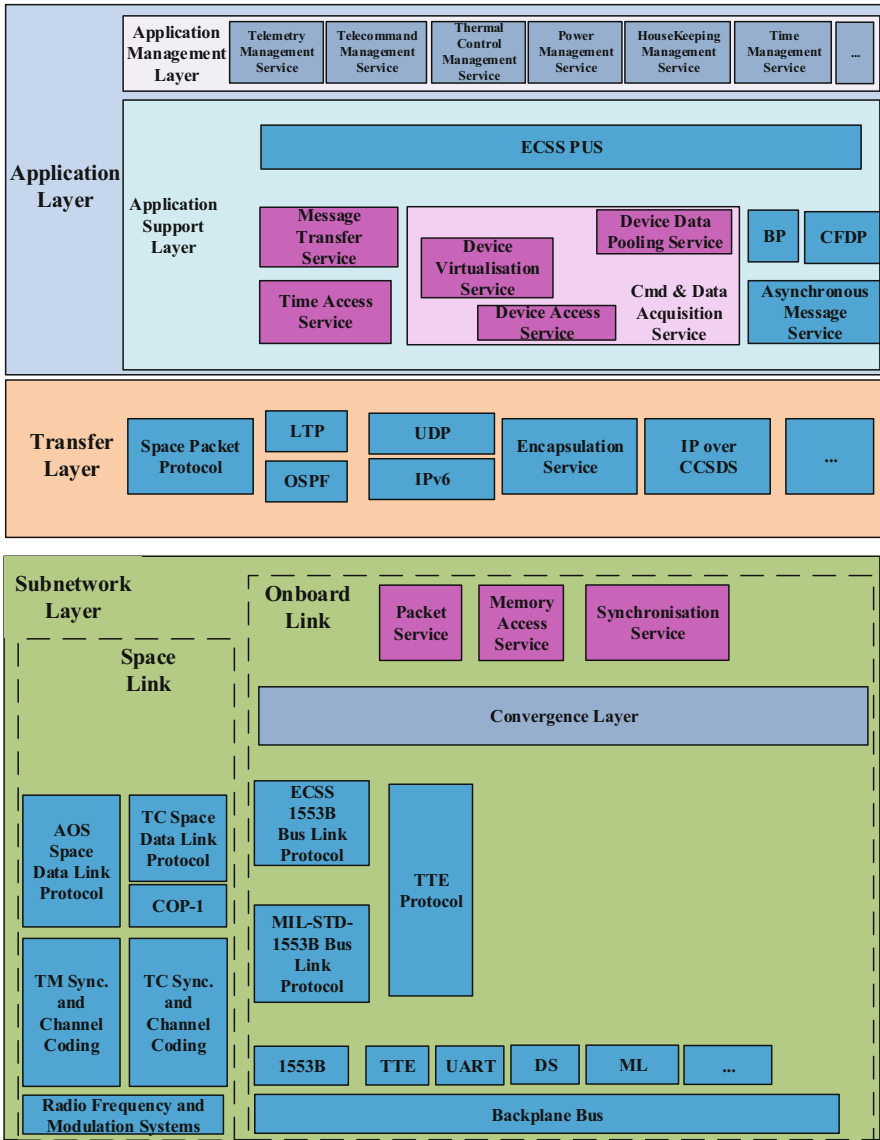


Fig. 2. The protocol architecture of avionics architectures for Software-Defined Spacecraft

management layer and extended applications can be implemented through the service combinations in the application support layer.

The hardware architecture is shown in Fig. 3.

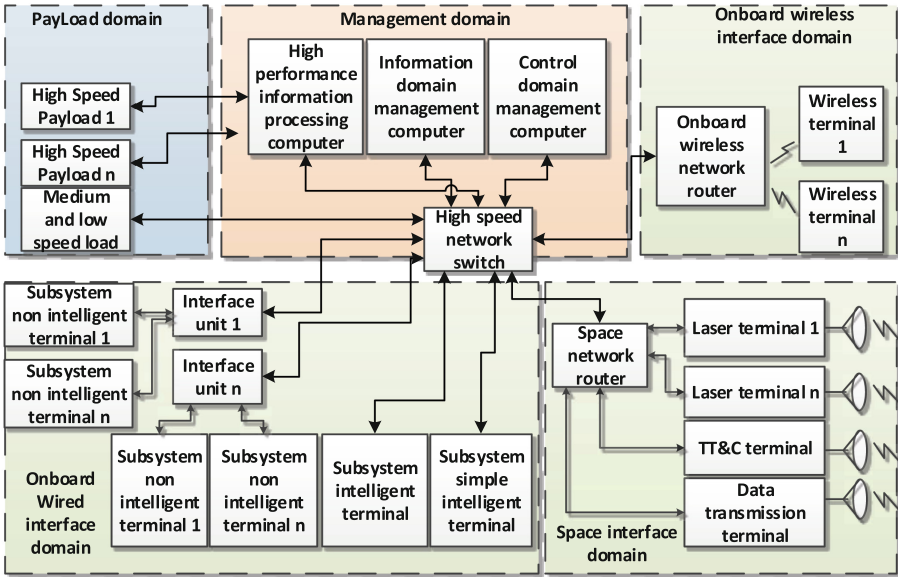


Fig. 3. The hardware architecture of avionics architectures for Software-Defined Spacecraft

The hardware architecture is divided according to the domain which are the management domain, wired interface domain, wireless interface domain, spatial interface domain and load domain. Each domain is interconnected through high-speed network switches, while the domain is interconnected through wired/wireless networks to jointly build an interconnected network.

Its main features of the proposed hardware architecture include:

- (1) Supporting functions migration: The computer of the system is a computing resource shared by each subsystem, and it can be interconnected with the equipment of this spacecraft/other spacecraft through high-speed network switches and space network routers, so that various functions can be migrated between different computers.
- (2) Supporting information fusion: The fusion of different types of information can be achieved through high information processing computers. For example, the fusion of optical information and SAR information can be achieved by remote sensing satellites to generate comprehensive intelligence.
- (3) Network definability: The internal network and the inter-device network can be defined by software. The network parameters, such as network topology, bandwidth and routing policy, can be configured as required.
- (4) Hardware reconfiguration: the reconfiguration in case of hardware failure can be achieved through the combination of modules, and the algorithm reconfiguration and upgrading of on-orbit hardware FPGA can also be completed through on-orbit loading.

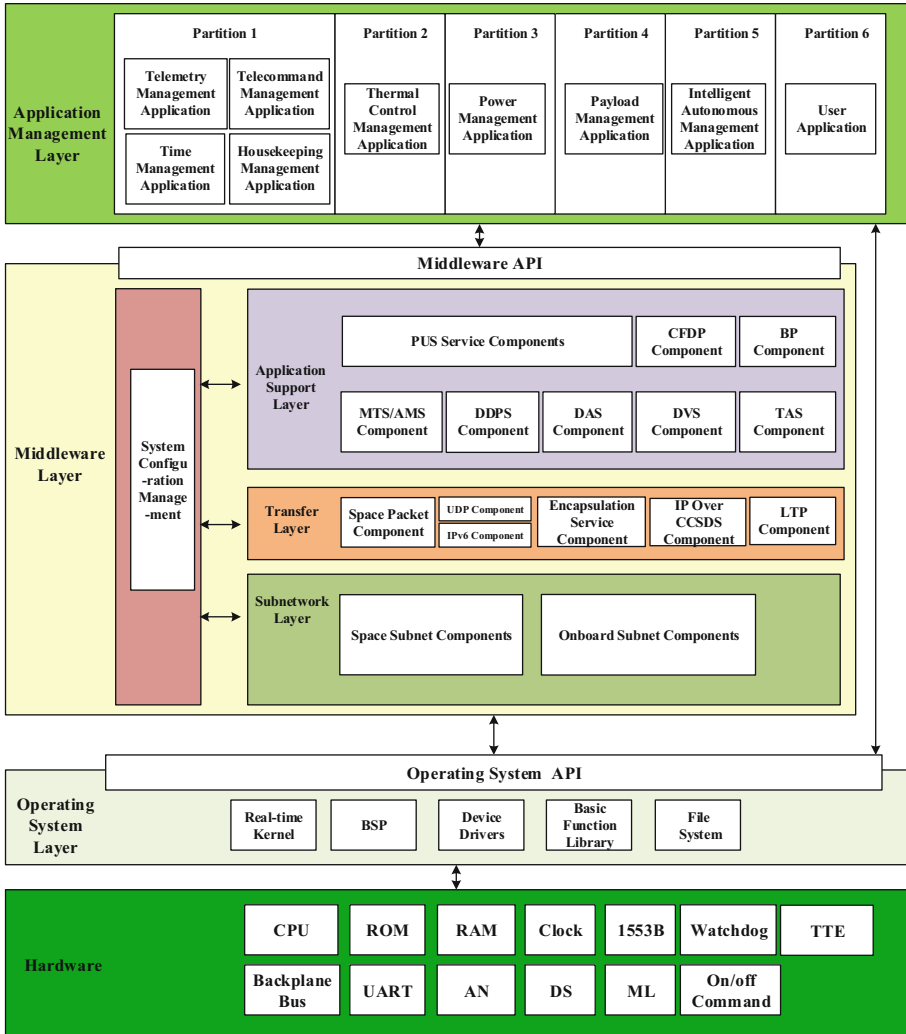


Fig. 4. The software architecture of avionics architectures for Software-Defined Spacecraft

- (5) Software loading-ability: the time-sharing and partition operating system is adopted to support the on-orbit loading of various software and to implement the expansion and upgrade of functions.

The software running in each computer/interface unit/intelligent terminal/space network router adopts a unified architecture, which is configured and tailored according to requirements. The corresponding software architecture is shown in Fig. 4. The above software architecture is mainly divided into the following parts:

- (1) Hardware layer: Hardware layer plays as the basis of software architecture, which includes components of various on-board computers such as CPU.
- (2) Operating system layer: Operating system layer is the underlying supportive layer of the software architecture, which encapsulates the interface of the operating system, provides a unified application programming interface for the operating system, and uses a time-sharing and partition operating system to support the safe and reliable operation of applications of different levels developed by different users on a unified processor.
- (3) Middleware layer: Middleware layer is the basic service layer of the software architecture, which realizes various standardized protocols and algorithms in the protocol architecture, and supports CCSDS, ECSS, IETF and other standard protocols.
- (4) Application layer: Application layer is a combination of multiple partitions. These partitions can not only realize the normal platform and load management functions, but also support the dynamic loading of applications from different users.

3.4 Key Technologies

Based on the requirements analysis and architecture proposed above. The main key technologies relevant to the avionics systems for Software-Defined Spacecraft are summarized as follows.

- (1) Heterogeneous distributed high performance computing technology on satellite.
Heterogeneous distributed high performance computing technology is necessary to develop high-performance multi-core computers and supporting management and scheduling algorithms to support the operation of multiple applications on the same computer. At the same time, to meet the requirements of high-performance computing such as image processing and information fusion, it is necessary to support the efficient collaboration between CPU, GPU, DSP, FPGA and other heterogeneous processing modules, as well as the collaborative processing requirements of computers between multiple spacecraft.
- (2) Software Defined Network Technology for Space Network
In order to meet the requirements of different spacecraft to reconstruct the network topology, customize or reconstruct the network routing, it is necessary to develop software definition network technology suitable for space networks, support the interconnection between heterogeneous spacecraft, support the on-demand change and expansion and upgrading of space network protocols, etc.
- (3) Dynamic loading and migration technology of space APP
In order to meet user-specific demand to load applications in on-board computers, it is necessary to break through the space APP dynamic loading technology, support lightweight online dynamic links, support the security isolation of different APPs, and support the dynamic migration of APPs between different computers.

(4) Space Software Middleware Technology

It is necessary to develop software middleware suitable for space, and provide unified interfaces such as device access, file access, plug-and-play, message sharing, reliable transmission, network management, time synchronization and operation synchronization for various APPs, making the development of APP regardless the underlying protocols and hardware.

4 Conclusion

In view of the requirements from multiple space application fields, this paper proposed architectures of avionics systems for Software-Defined Spacecraft and sorts out the key technologies. The proposed architectures consists of protocol architecture, hardware architecture and software architecture. The adoption of avionics systems for Software-Defined Spacecraft will bring innovations to spacecraft design concept and design method, which can be conducive to the realization of mass production, function on-orbit definition, spatial networking, and autonomous intelligence of the next generation of spacecraft. The key technologies proposed in this paper still need further research and breakthrough.

References

1. Zhou, Y., Zhang, X., Zhi, M.Y.: Research on software defined satellite payload technology. *J. Northwestern Polytechn. Univ.* **38**(S1), 96–101 (2020)
2. Walker, J.L., Mckinnon, D.: Future digital flexible and software defined payload systems for commercial space. In: *AIAA International Communications Satellite Systems Conference and Exhibition*, pp. 2–13. Queensland, Australia (2015). <https://doi.org/10.2514/6.2015-4325>
3. Li, W., Chen, J.: Latest progress of European and American software defined satellite project. *Int. Space.* **1**, 53–55 (2020)
4. Yang, X.: Concept research on “Software satellite”. *Electron. Warfare.* **1**, 1–5 (2002)
5. Zhao, J., Wu, F., Liu, G., et al.: Systematic idea and technical practice of developing software defined satellite. *Sat. Netw.* **4**, 44–49 (2018)
6. He, X., Li, N., Xu, Y., et al.: Requirements analysis of intelligent spacecraft avionics system and discussion of its architecture. *Spacecr. Eng.* **27**(4), 82–89 (2018)
7. Baggerman, C., McCabe, M., Verma, D.: Avionics system architecture for the NASA orion vehicle. In: *SAE AeroTech Congress and Exhibition*, pp. 1–10, Seattle, the United States (2009). <https://doi.org/10.4271/2009-01-3276>
8. Black R., Fletcher M.: Next generation space avionics: layered system implementation. *IEEE Aerosp. Electron. Syst. Mag.* **20**(12), 9–14 (2005)
9. Cheng, B., Liu, W., He, X., et al.: Research on Orion electronic system. *Spacecr. Eng.* **25**(4), 102–107 (2016)
10. He, X., Guo, J., Li, Y.: Autonomous health management requirements and software architecture for deep space probe. *Control Theory Appl.* **36**(12), 2065–2073 (2019)
11. Feng, X., Li, Y., Yang, C., et al.: Structural design and autonomous mission planning method of deep space exploration spacecraft for autonomous operation. *Control Theory Appl.* **36**(12), 2035–2041 (2019)

12. Xu, Y., Li, J., Lin, X., et al.: An optimal bit-rate allocation algorithm to improve transmission efficiency of images in deep space exploration. *China Commun.* **17**(07), 94–100 (2020)
13. CCSDS 850.0-G-2. Spacecraft onboard interface services. CCSDS, Washington D.C. (2013)