Chapter 48
Product Line Architecture for TT&C Software Support Framework

Zhilin Wang, Xinming Li and Dongdong Li

Abstract One of the most promising techniques to improve the quality and productivity of TT&C software is product line engineering, the core of which is the product line architecture for TT&C software support framework (PLATSSF) which is of great help to improve the reusability of architectural design. Since component models are cornerstones of support framework design, the characteristics of them including variety, common requirement and variant requirement were firstly analyzed. Then, mainly by analyzing the three stages of TT&C software, which include development deployment and runtime, the structure of PLATSSF is designed. Thirdly, the function of PLATSSF is instantiated by adding a new satellite to the framework. In this instance the software requirements are analyzed and the contents of software development are provided. In conclusion, PLATSSF implements the production of the TT&C software series, reduces the workload of software development and improves the software quality.

Keywords TT&C software · Software product line architecture · Software support framework · Component model

48.1 Introduction

Space TT&C Network is a special network by which orbited spacecrafts are tracked, measured and controlled [1]. With the expanse of the space enterprise, space TT&C center and the space application center need highly cooperate with

Z. Wang (✉) · D. Li
Beijing Space Information Relay and Transmission Technology Research Center, Beijing 100094, China
e-mail: zhilinwww@126.com

Z. Wang · X. Li
Science and Technology on Complex Electronic System Simulation Laboratory, Beijing 100092, China

R. Shen and W. Qian (eds.), *Proceedings of the 27th Conference of Spacecraft TT&C Technology in China*, Lecture Notes in Electrical Engineering 323,
DOI 10.1007/978-3-662-44687-4_48

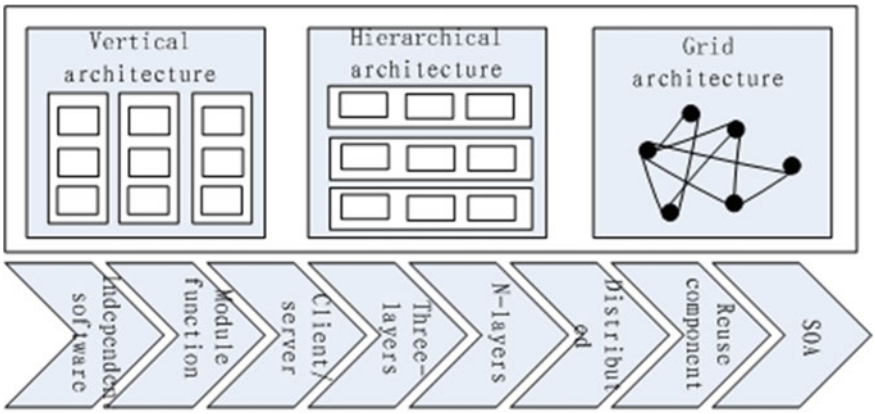

Fig. 48.1 Evolvement of TT&C center software architecture

each other to complete the complex tasks. Space TT&C center is the main control node of the TT&C network, in which TT&C software plays a key role to automate and intellectualize the system. Software support framework provides a deploying and running environment for the function components, and is foundation of space TT&C center software. With increase of mission requirements, TT&C center software's architecture is making the transit to the network one. All in all, software support framework is now facing with new challenge.

In the past 10 years, TT&C center software architecture went through three phases. The first one is vertical architecture, including many representations, such as independent software, module function and client/server. The second one is hierarchical architecture, including many representations, such as client/server, three-layers, N-layers and distributed structure. The Last one is grid architecture, including many representations, such as distributed structure, reused component and SOA. As Shown in Fig. 48.1.

By Far, hierarchical architecture is mainstream and popular for TT&C center software. With increase of requirements, some of TT&C center software is adopting grid architecture.

Component-Based Development (CBD) [2] builds software system by combining the developed components, which can reduce development cost, and improve reuse rate of developed components [3].The developed components are the base units of combination, describe the offered service by interface, define the demand for environment, and can be independently deployed by a third party.

The support framework provides the running and deployable environment for components [4], and is fundamental of the component-based distributed development. During the running state, support framework builds and manages instances of components, allocates system resources, maintains intercommunications of components, and transparently heads off request of the outer system. As support framework has achieved common design and implementation [5], developers only need to focus on certain parts, without considering common parts.

Product line engineering is a software engineering technology which develops special domain product series by using common kernel assets [6]. It involves three phase: domain analysis, domain design and domain completion. Domain analysis is different with common requirement analysis, adds description for commonness and variability of the special domain, and guides the manufacture of reused software assets. Product line architecture (PLA) [7] is a product of domain design. By supporting the domain requirement and implementing the common requirement, PLA realizes systematic reuse. Research on PLA can promote structural reuse of support framework, and improve productivity and quality.

48.2 Domain Analysis of Support Framework

48.2.1 Component Sort

Considering universal principle of function definition in the field of TT&C, TT&C system falls into two categories: task function component and system manage component. Task function component satisfies requirements of special tasks, including compute component, satellite control component, equipment control component, customer data relay component and customer data parse component. System manage component satisfies requirements of system support framework, including task layout component, task schedule component and status surveillance component. The following lists are the details of each component function:

1. Compute component. It receives and deals with the trajectory data, and computes the orbits and the control coefficients of satellites.
2. Satellite control component. It implements the TT&C operations of the satellites.
3. Equipment control component. It remotely monitors and controls of the ground stations.
4. Customer data relay component. It exchanges customer data customer data between the ground stations and the customer centers.
5. Customer data parse component. It deals with data of satellites.
6. Task layout component. It allocates the resources of Space TT&C Network, establishes and manages the task layout.
7. Task schedule component. According to the layout or human instruction, it schedules the all function components of the software system.
8. Status surveillance component. It shows the task progress and system status.

Relation among aforementioned components is described in Fig. 48.2.

Liu Guo-Liang [8] details the analysis framework of the component model. The author aims at the software system of space TT&C.

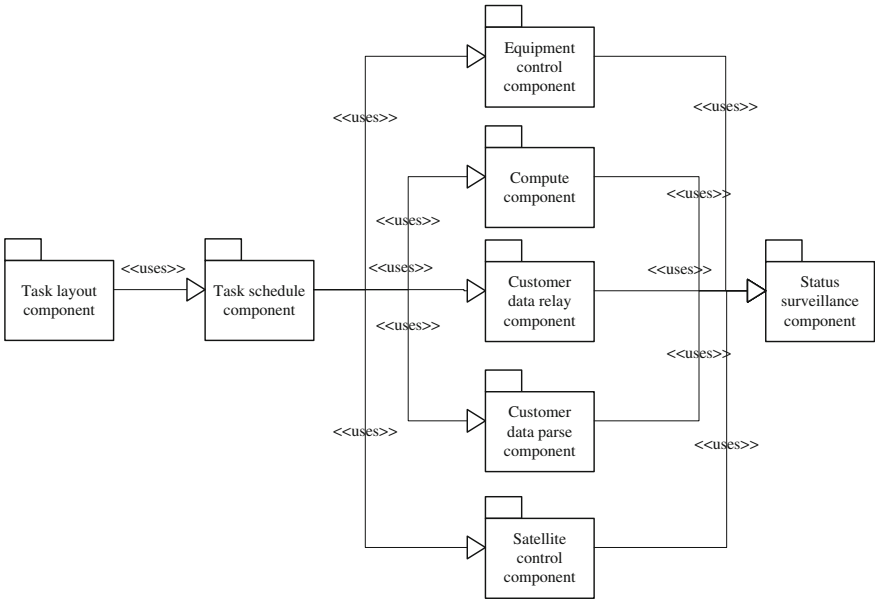

Fig. 48.2 Relation among function components

48.2.2 Component Common Requirement Analysis

The common requirements of component support framework include component deployment, component running, and component maintenance. There are five actors associated with component support framework, which are deployer, manager, operator, resources, and component instance. By analyzing relation among actors and the support framework, domain case model is educed as shown in Fig. 48.3.

By decomposing cases into a series of independent acts, it is achieved that the function requirements refine. In outer view, each act is indivisible. For example, the task cooperation case represents components' cooperates in task. It includes scheduling synchronization, data synchronization, and operation synchronization.

48.2.3 Component Variant Requirement Analysis

There are five sorts of variant requirements, which are symbol configuration, script template, network configuration, display configuration, and special algorithm. As shown in Table 48.1.

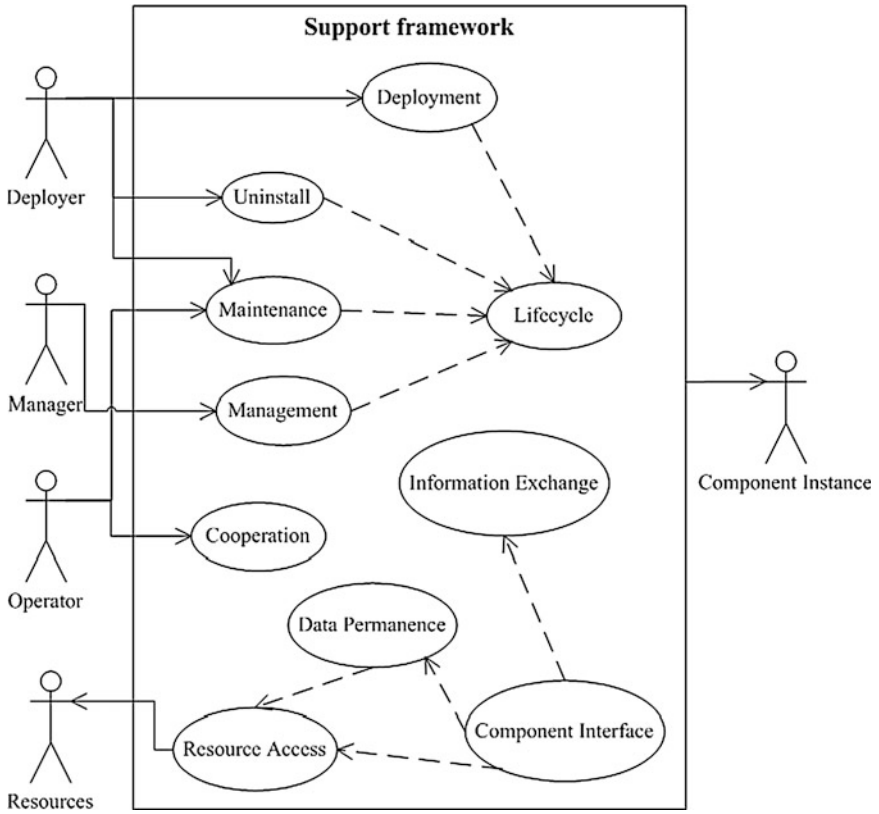

Fig. 48.3 Domain case model

48.3 Product Line Architecture for TT&C Software Support Framework

48.3.1 Development Structure

The development structure includes two parts. See Fig. 48.4.

1. Basic software layer. It contains support framework and component set. It mainly solves the problem of the system common function design, and builds basic platform for task expanse. The edition of basic software is independent with task.
2. Task software layer. Aiming at the special requirements of tasks, it is developed based on basic software layer. It mainly contains various configuration files and some programs for special algorithm. Each task software is independent.

Table 48.1 Analysis of component variant requirement

Type	Description	Software structure	Application
Symbol configuration	There are much similar algorithms in the different tasks. Only by adjusting several arguments, the same algorithm can be reused	The structure of argument set is adopted. By adding new configuration options, the new tasks' requirements are satisfied. By choosing the "file + directory" as the form of argument set, the chief file appoints the configuration directory. The configuration directory includes many files. Each file only contains the requirements of one task	Compute component Task layout component
Script template	The procedure is mostly consistent in different tasks. The main difference is the amount and values of arguments	The structure of "script + argument set" is adopted. The script is collection of procedure and argument symbol. The argument set is a list of argument symbols and values. In running time, script and argument set are dynamically combined. In order to satisfy the different types of tasks, different scripts are applied	Satellite control component Equipment control component Task schedule component
Network configuration	The basic function of data exchange for all customers is same. The differences include IP address, amount of link, and the exchange rule of application layer	The structure of "domain name service + route algorithm" is adopted. The domain name service achieves the mapping between IP addresses and customers. The route algorithm implements the exchange rule of application layer	Customer data relay component
Display configuration	The display style is much different for each task. But the basic display elements are same	The structure of "display framework + page deployment" is adopted. The display framework provides the basic display elements. The page deployment defines the combination of the display elements	Status surveillance component
Special algorithm	There is completely different in customer data. Special algorithm need to developed	The structure of "software algorithm framework + special program + compatible interface" is adopted. The software algorithm framework and special program insure that each special algorithm is developed independently. The compatible interface insures that the data format is consistent	Compute component Customer data parse component

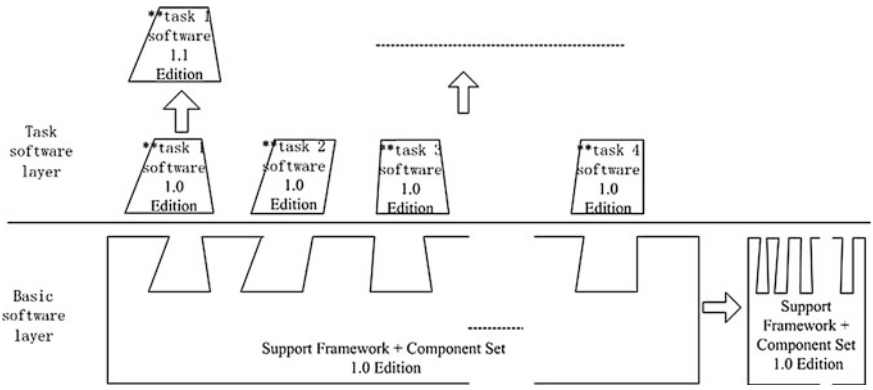

Fig. 48.4 Development structure

Under this development structure, maintenance of basic software and task software can be simultaneously implemented, and different editions of softwares are managed and upgraded respectively.

48.3.2 Layer Structure

The analysis of variant requirement is the base of the component reuse. There are four layers for the component reuse. See Fig. 48.5.

1. Task Procedure. A series of acts make up of one task procedure. Each act matches one component.
2. Component support framework. It provides the running environment for component.
3. Component module. It is component set, and in.
4. Component configuration set. It includes series of configuration files.

Every Task Procedure uses the command interface to send instruction to the component support framework. The component support framework parses the instruction and uses the module loading interface to initialize component. The component module uses the file configuring interface to get task arguments, so as to configure itself at initialization time.

48.3.3 Running Structure

Based on support framework, task software implements the unity among development, deployment, and maintenance. However, in running structure, each task software is independent. According to the task requirements, the editions of

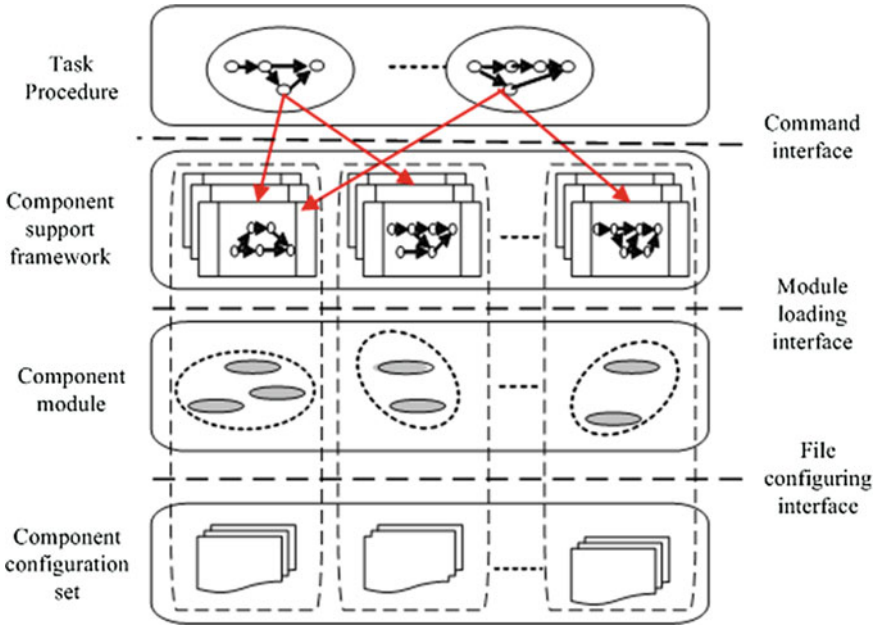

Fig. 48.5 Layer structure

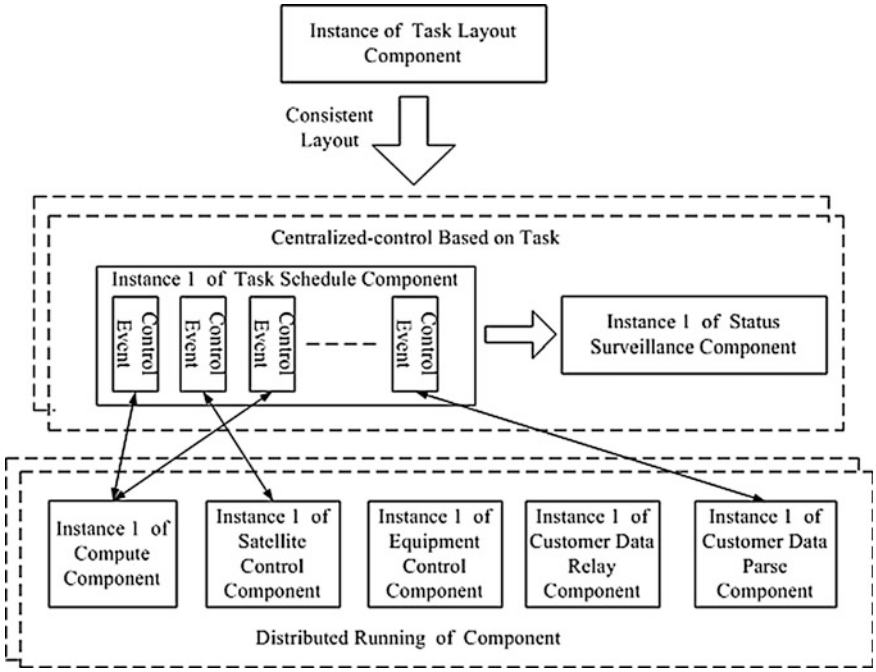

Fig. 48.6 Running structure

configuration files are finished. By configuration files, we initialize the instances of the requisite components. See Fig. 48.6.

In the system runtime, there are three kinds of initialization modes for all component instances. Firstly, it's task layout component which initializes only one instance in the whole system. This instance provides service for all customers and confirms the exclusive result of resource layout. Secondly, they're task schedule component and status surveillance component, which initialize one instance for each task. They build the independent platform for special task to be monitored. At last, all other components are initialized on demand. Every instance is independently running and focus on single function.

In a word, ultimate purposes are achieved, including consistent layout, centralized-control based on task and distributed running of component.

48.4 Software Product

By adding a new satellite to space TT&C Network, we validate the effect of PLATSSF. The task requirements of the satellite have two ways. The one is providing management of satellite platform from orbiting, which includes three stages: going into orbit, on-orbit, seceding orbit. The other one is providing service of data relay.

48.4.1 Task Analysis

48.4.1.1 Management of Satellite Platform

The task requirements include position keep, attitude control, power management, and orbit calculation etc. We take more complex position keep as case and analyze the task requirements. See Table 48.2.

48.4.1.2 Service of Data Relay

Through operating satellite and station equipment, we build a data link and implement application data exchange between customer and satellite. See Table 48.3.

Table 48.2 TT&C task analysis

Num	Require of task control	Phase	Operation instance
1	Orbit prediction	Task layout	Software process
2	TT&C task application		Software process
3	Station equipment prepared	Task running	Station equipment
4	Bidirectional capture control		Station equipment
5	Telecontrol channel examine		Satellite
6	Control variable compute		Software process
7	Instruction inject		Satellite
8	Task implement		Software process
9	Procedure surveillance		Software process
10	Orbit calculate	Task evaluation	Software process
11	Control effect analysis		Software process

Table 48.3 Data relay task analysis

Num	Require of task control	Phase	Operation instance
1	Orbit prediction	Task layout	Software process
2	Data relay task application		Software process
3	Station equipment prepared	Task running	Station equipment
4	Data relay equipment prepared		Station equipment
5	Bidirectional capture control		Satellite
6	Telecontrol channel examine		Satellite
7	Instruction inject		Software process
8	Customer data relay		Station equipment
9	Customer data parse		Software process
10	Procedure surveillance		Software process
11	Task implement		Software process
12	Information exchange	Other	Software process

48.4.2 Maintenance Requirement and Product Format

According to the function of all components, we divide task requirements into sixteen parts, get the maintenance requirement of the software, and design the product modes for every function. See Table 48.4.

Through combining the above software products, we get the special software product line of the TT&C center, and make the reuse rate of the software greatly improved. See Table 48.5.

Table 48.4 Maintenance requirement and product format

Num	Component	Require of task control	Maintenance requirement	Product mode
1	Compute component	Orbit prediction	Adding satellite symbol	Argument configuration file
2		Orbit calculate		
3		Control variable compute	Adding satellite constants and control requirement	
4		Control effect analysis		
5	Satellite control component	Telecontrol channel examine	Adding satellite instruction sequence and telemetering description	Satellite telecommand script
6		Instruction inject		
7	Equipment control component	Station equipment prepared	Adding control command set	Equipment control command
8		Bidirectional capture control		
9		Data relay equipment prepared		
10		Customer data relay		
11	Customer data relay component	Information exchange	Adding network configuration	Route configuration file
12	Customer data parse component	Customer data parse	Adding customer data format description	Data parse configuration file
13	Task layout component	TT&C Task application	Adding satellite symbol	Argument configuration file
14		Data relay task application		
15	Task schedule component	Task implement	Adding task procedure	Task schedule script
16	Status surveillance component	Procedure surveillance	Adding display pages	Page configuration files

Table 48.5 Summarization of software reuse

Task type	Maintenance type	Reuse rate (%)
The sequence kind one of current spaceflight	Configuration files	95
New spaceflight, but task requirements is similar	Configuration files + special algorithm	80
New spaceflight, but task requirements is different	Configuration files + special algorithm + new component	70

48.5 Conclusion

According to theories of software product line and support framework, software requirement of the TT&C domain is analyzed, including component sort, common requirement and variant requirement. Product line architecture is designed for TT&C Software support framework. From three views of development structure, layer structure, and running structure, Systematic structure is described in detail to achieve a software system based on the support framework. Aiming at adding a new satellite into TT&C network, task requirements are analyzed, maintenance requirement and product format of software system are demonstrated, and the reuse rate of software systems is simply summarized.

References

1. Hao Y (2004) TT&C network. China national defense industry press, Beijing
2. Wang YH (2009) Software component and architecture. China machine press, Beijing
3. Zhang W, Mei H (2003) A feature-oriented domain model and its modeling process. J Softw 14 (8):1345–1356
4. Bachmann F, Bass L, Buhman C et al (2000) Volume II: technical concepts of component-based software engineering. In: Technical report, 2nd edn CMU/SEI-2000-TR-008, Software Engineering Institute, Carnegie Mellon University, Pittsburgh
5. Mei H, Chen F et al (2003) ABC: an architecture based, component oriented approach to software development. J Softw 14(4):721–732
6. Clements P, Northrop L (2002) Software product line: practices and patterns. Addison-Wesley Professional, Boston
7. Bosch J (2000) Design and use of software architectures: adopting and evolving a product-line approach. Addison-Wesley Professional, Reading
8. Liu GL, Wei J (2010) Container product line architecture based on component model analysis. J Softw 21(1)