Real-time Data Process Software for POAC Space Mission Management System

Yi Qu[1], Xuzhi Li[2], Yurong Liu[3], Juan Meng[4], Jun Rao[5]
General Establishment of Space Science and Application, Chinese Academy of Sciences1-5, Beijing, 100094, China

This paper gives a description for the real-time data process software for POAC space mission management system, including its function requirement, performance requirement, interfaces with other software in the system, architecture design and process flow. The software uses multiple processes architecture to meet function requirement and achieve satisfactory performance. The processes run in parallel and communicate through data queues. The software has supported several missions, proved to be effective and easy to adapt heterogeneous missions.

Nomenclature

APID = Application Identification
DAP = Data Archiving Process (One of RDP's working processes)
DCP = Data Calculating Process (One of RDP's working processes)
DRPEP = Data Receiving and Package Extracting Process (One of RDP's working processes)
DRT = Data Receiving and Transformation (Software of POAC's mission management system)
FD = Fault Diagnosis (Software of POAC's mission management system)
LP = Log Process (One of RDP's working processes)
OC = Orbit Calculation (Software of POAC's mission management system)
OIM = Operation Information Management (Software of POAC's mission management system)
POAC = Payload Operation and Application Center
PWS = Payload Work Schedule (Software of POAC's mission management system)
RDP = Real-time Data Process (Software of POAC's mission management system)
RSM = Real-time Status Monitor (Software of POAC's mission management system)

I. Introduction

POAC (Payload Operation and Application Center, General Establishment of Space Science and Application，Chinese Academy of Sciences) acts as the ground segment for many space missions, such as Double Star Project, HJ Satellite Constellation, SJ Satellite and so on. It organizes integrated payload tests and science experiments in-orbit, allocates space-based and ground-based resources, manages and controls payloads, provides data products to users for long term.

[1] Engineer, Payload Operation and Application Center (POAC), quyi@csu.ac.cn.
[2] Professor, Payload Operation and Application Center (POAC), xzhli@csu.ac.cn.
[3] Associate Professor, Payload Operation and Application Center (POAC), yurl@csu.ac.cn.
[4] Senior Engineer, Payload Operation and Application Center (POAC), mj@csu.ac.cn.
[5] Senior Engineer, Payload Operation and Application Center (POAC), raojun@csu.ac.cn.

POAC space mission management system consists of many software, hardware, devices and staff. This paper focuses on its Real-time Data Process (RDP) software, which is one of the basic software of POAC space mission management system.

Figure1 shows the general data process flow of POAC space mission management system. When data are downloaded from spacecrafts, they are transferred to POAC from tracking station in real-time. After simple recording and formatting by Data Receiving and Transformation (DRT) software, the data are transmitted to RDP by LAN. RDP analyzes the structure of the data, gets their frame type, extracts data packages from frames and calculates every parameter in the packages to meaningful and readable results. Then RDP sends results to other software in the system to display，to make further analysis or to support other application. From these results, experts and operators can monitor payloads' work procedure, estimate their health status, and make operation plan for the next stage.

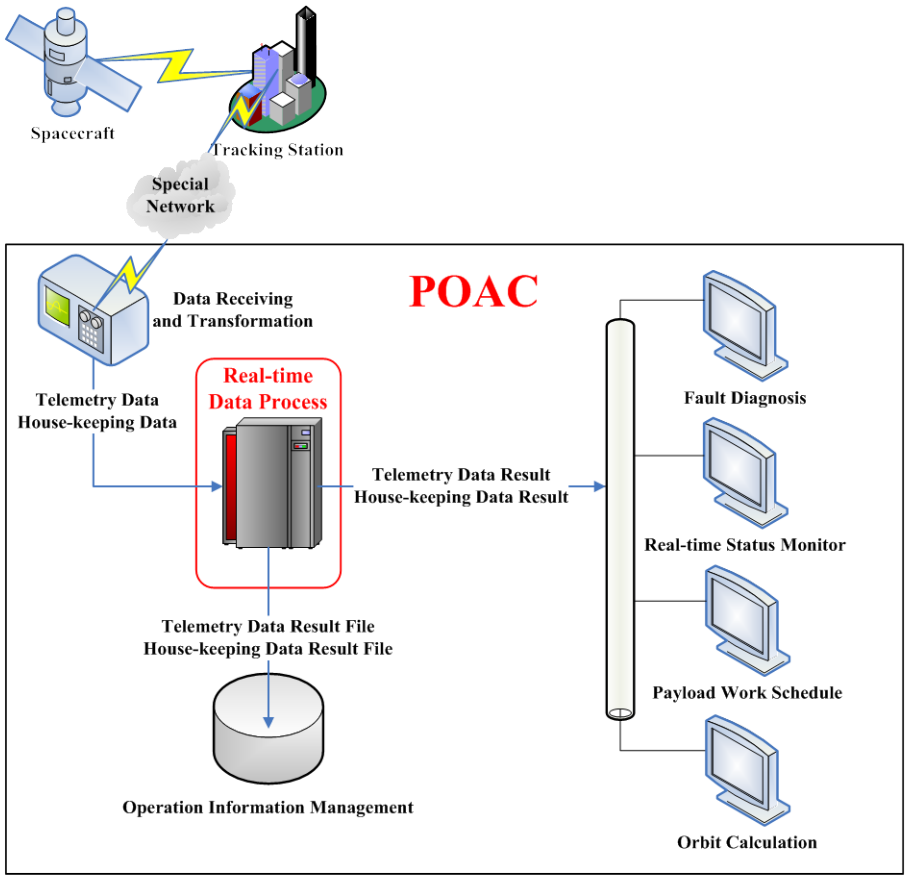

Figure1 General Data Process Flow of POAC Space Mission Management System

II. Function Requirement

The functions that RDP should accomplish include:

Data receiving and data frame verification: DRT sends data to RDP by LAN, and RDP must receive the data by special network address and port. Once RDP gets data from network, RDP will check the data format and their content according to data transfer protocol. If not valid, RDP will discard the data to protect itself.

Data package extraction: After verification, RDP extracts data packages from data frames with APID and other identifications in the packages. If the package is longer than current frame's data area can hold, RDP must buffer the data until a complete package is obtained or later data are proved to be other package because of poor data quality.

Primary data statistics: RDP should make a primary statistics to data, including receiving start time, receiving end time, frame count, package count, invalid frame count, original data amount, result data amount and so on. This may help to long term running analysis, especially when fault happens.

Parameter calculation: When getting a complete data package, RDP will pick out the binary data for every parameter from package's data area, and put the binary data into meaningful and readable result according to parameter's algorithm. Since there may be hundreds of parameters in a package, this course may cost much time.

Parameter result organization and broadcast: RDP will not send out the parameter result one by one which will decrease the performance of network. Instead it will assemble all parameter results in one package into a result string and send out the result string once per package.

Parameter result archive: RDP should make parameter result files to support advanced data process and analysis. There are many types of packages, and each type may have many classes, so RDP should make archive according to users' requirement. Besides, RDP should design file manage rule carefully to make files easy to search and avoid naming collision.

Disk space check: RDP should check available disk space regularly. If available space is smaller than pre-assigned threshold, RDP should warn operators and mangers to prevent data lost. Furthermore, RDP should control the size of each file to ensure not bigger than operation system and database can tolerate.

Work log record and sending: RDP should keep work log for its whole run time to provide convenience to future analysis. Except sending log information to remote database in LAN, RDP should keep the same local one as a backup.

III. Performance Requirement

Except the functions listed above, RDP needs a high performance to adapt the rapid development of aerospace technology.

Reliability: Many phenomena can affect aerospace tracking, and data errors can occur during the transfer from space to ground unexpectedly. RDP must take actions to avoid software crash down when facing data errors. What's more, RDP should simplify its algorithm to decrease the probability of software bug.

Stability: Current space mission's run time has increased greatly than before, many of them can run several years or longer. As a service on the ground, RDP's run time must match space mission's run time. This means RDP must allocate memory and other resources carefully to maintain a relatively stable running environment. Also RDP should distribute the workload reasonably, and try to keep its balance.

Compatibility: Though all departments have realized the importance of standardization, heterogeneous missions still exist. RDP should set up a flexible architecture to support more missions other than customization for a special mission. Software modules should be divided reasonably and interface among modules should be standard so that the software is easy to extend.

Rapidity: Many operators and experts need to acquire spacecraft and payloads' status as soon as possible, so RDP should try its best to meet the time requirement.

IV. Interfaces with Other Software

A number of software in POAC mission management system has interface with RDP, some of them sends data and message to RDP and others receives RDP's data result and message, as Fig. 2 shows.

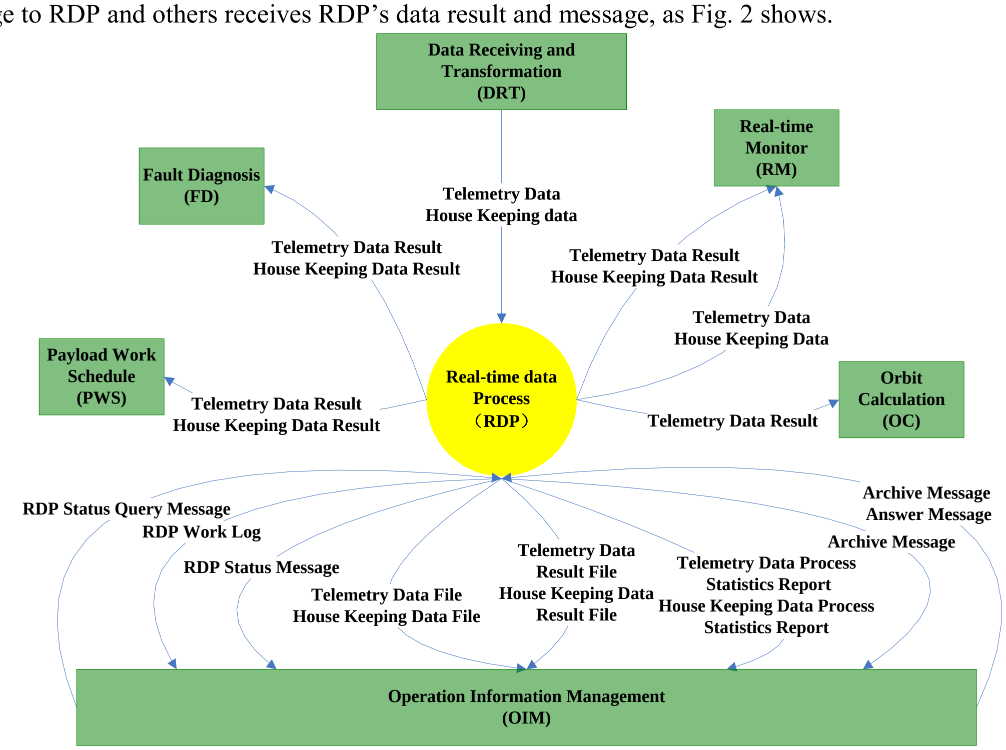

Figure 2 Interfaces with Other Software

The interface between RDP and DRT is RDP's main input. DRT sends telemetry data and housekeeping data to RDP in real-time, whereas RDP needs not give DRT feedback. DRT and RDP make a data transfer protocol in advance, which describes the data structure, data transmitting method, network setting and other information for each other. Then both of the software will design according to this protocol. Now the interface is realized by point-to-point network in POAC mission management system.

The interface between RDP and Real-time Status Monitor (RSM) is RDP's main output. RDP will give RSM telemetry data, telemetry data result, housekeeping data and housekeeping data result, also RSM needs not give RDP feedback, RSM will display these data to operators and experts. All of the data RDP provided to RSM is text, and RSM needs not any calculation or format translation to display. Since there are many terminals, RDP realizes the interface by broadcasting data in LAN.

The interfaces between RDP and Orbit Calculation (OC), Payload Work Schedule (PWS), Fault Diagnosis (FD) are similar to the interface between RDP and RSM. The difference is these software receive less data than RSM, they only receive special data they needed from network. When getting needed data, they may transform the data format according to their own algorithm, and then they make further analysis or other application based on the data. Also these interfaces are realized by broadcast.

The interface between RDP and Operation Information Management (OIM) is mainly about file archive and status message. RDP will not deal with database directly, instead it will send all its files to OIM, and inform OIM with an archive application message. When OIM gets the archive application message, it will accept all the files and put the file information into database, then it will send archive answer message to RDP saying the archive has finished. Besides, OIM will collect information from all software in POAC mission management system to estimate their running status. So it sends status query message to RDP in a fixed frequency, and RDP will respond t it with a status message. What's more, OIM also collects work log from other software to put into log database, so RDP sends work log to OIM when producing log. The interface is realized by broadcast to simplify OIM's work flow.

V. Architecture

RDP distributes the function requirement into four independent processes, which are named as Data Receiving and Package Extracting Process (DRPEP), Data Calculating Process (DCP), Data Archiving Process (DAP) and Log Process (LP).

Expressed with the function items listed in Part II, DRPEP answers for data receiving and data frame verification, data package extraction; DCP answers for parameter calculation, parameter result organization and broadcast; DAP answers for parameter result archive, primary data statistics and disk space check; LP answers for work log record and sending.

To facilitate inter-process communication, RDP creates four data queues, which are named as Data Package Queue, Data Result Queue, Statistics Queue and Log Queue. They act as interfaces among processes listed above, and can buffer data if needed, as Fig. 3 shows.

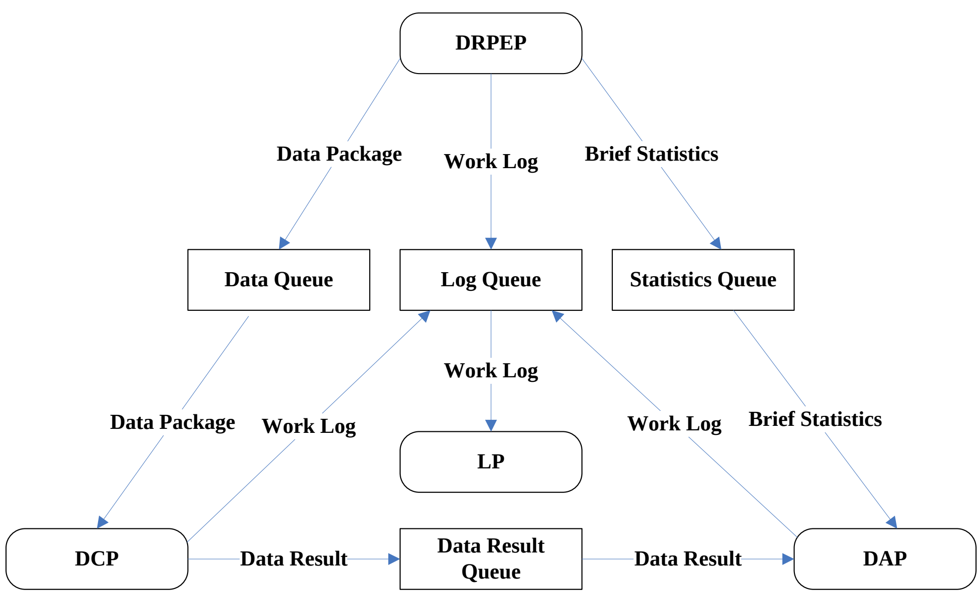

Figure 3 Process Communications using Data Queues

The content in Data Package Queue is binary data packages which DRPEP pick out from data frames. These data packages are entire packages, with parameters and identifications. Data package queue is written by DRPEP, and read by DCP.

The content in Data Result Queue is telemetry data result and house-keeping data result. All the results are readable text, part of the text is identifications, and the rest is parameter results. The queue is written by DCP, and read by DAP.

The content in Statistics Queue is brief statistics provided by DRPEP. They are binary identifications and other useful information picked out from data frames and data packages, such as time, tracking station, length, etc. They are not real statistics, just material to make statistics. The queue is written by DRPEP, and read by DAP.

The content in Log Queue is processes' work log, they are text. The queue is written by DRPEP, DCP and DAP, read by LP.

These queues are proprietary to each mission, which mean that different missions and different data types will use different data queues. This ensures each queue filled with similar data structure, which can simplify the procedure of reading and writing. Meanwhile, this can distribute data, avoid data queue overflow, and reduce queue management cost. As to the capacity of each queue, they must refer to the data rate and data amount of each mission. Instant data peak and long-term data deposit must be taken into account, and proper margin is necessary.

DRPEP, DCP, DAP and LP run independently, and communicate by data queues. The queues are created and managed by an independent process other than the processes listed above. The processes listed above can only read data from or write data into the specified queues, and they must obey the permission management and data protection rule of the queues. This can enhance the security of the queues, and make them alive even if any process listed above falls into trouble.

The architecture has many advantages. Firstly, it can simplify each process, decrease error or exception probability. Secondly, it confines the exception influence, which can reduce data loss which exceptions bring out. Thirdly, it permits all processes run in parallel, improving the time performance. Fourthly, it can be deployed in a distributed environment.

To adapt heterogeneous missions, RDP creates configuration files to describe information which is necessary to data process but customized for different missions. All configuration files have hierarchical structure, describing frame structure, package structure and parameter algorithm respectively. These files take XML Telemetric and Command Exchange (XTCE) as reference which CCSDS has submitted as a standard description for telemetry and telecommand data[1-2]. Whereas the XTCE schema is complex, and has some ambiguous information, RDP simplifies it greatly, just keeping those necessary to data process[3-4]. When supporting different missions, RDP creates a new set of configuration files and deploys with software programs together. This can reduce repeated development and keep software architecture stable.

VI. Process Flow

When RDP starts, it starts four processes in fact. LP, DAP, DCP and DRPEP start in sequence. After these processes start, they are resident in memory and run in background driven by data streams.

When spacecrafts enter tracking area, DRPEP receives data frames in real-time, validates their format and content, extracts data packages from frames, and puts data packages into data queue, puts brief statistics into statistics queue. DCP gets data packages from data queue in order, calculates all parameters according to their algorithms, integrates all parameter results into result strings and finally broadcasts the result strings in LAN, meanwhile putting the result strings into data results queue. DAP reads out result strings from data result queue, then writes them into their respective files, then DAP reads out brief statistics from statistics queue, and updates inner counters based on these statistics.

When spacecrafts leave tracking area, DAP renames all result files which are produced during the last tracking time according to the file name rule, generates statistics reports for the last tracking time, then creates archive for these files.

LP reads out work log from log queue regularly, records the information and sends out to OIM in LAN.

All these processes run independently in parallel, and communicate by data queues. Figure 4 shows the process cooperative work flow.

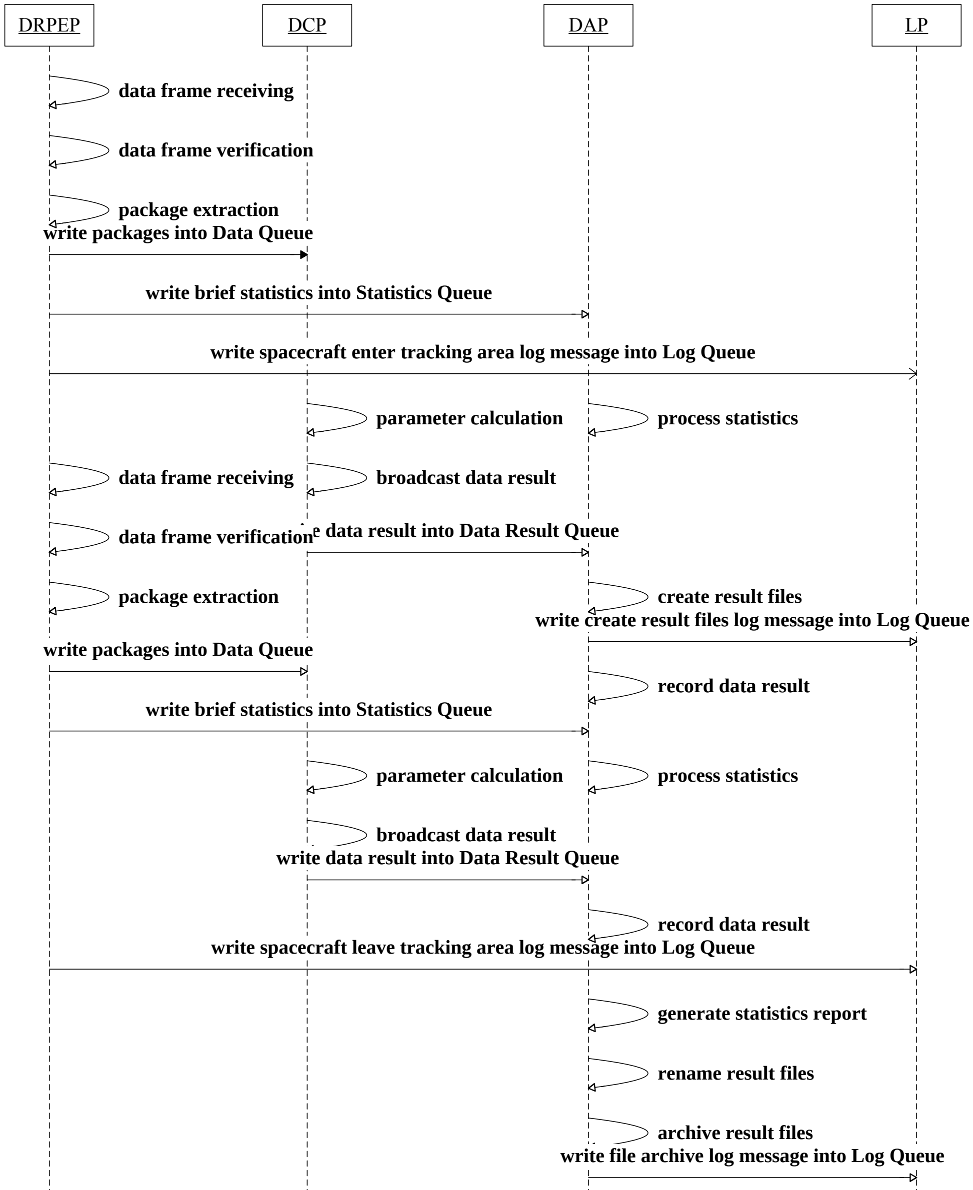

Figure 4 RDP Process Flow

VII. Conclusion

It is easy to realize RDP's function requirement in the test environment, but need more consideration to maintain a high performance during long-term mission operations. POAC has supported many space missions successfully, and RDP provided data result to users in real-time. In the recent mission, RDP receives and processes data from more than 50 types of packages, and has run effectively for more than half a year. As to the future, data rate and data amount will increase rapidly, data types will also increase greatly, so RDP needs more advanced technique to meet the challenge. Though devices and hardware develop very fast, software must improve its structure and algorithm to achieve success.

References

[1]CCSDS Secretariat, "CCSDS 660.0-R-2-2005 XML TELEMETRIC AND Command Exchange", Washington: CCSDS Secretariat, 2005

[2]CCSDS Secretariat, "CCSDS 660.0-B-1-2007 XML TELEMETRIC AND Command Exchange", Washington: CCSDS Secretariat, 2007

[3]LiuFeifei, "Space Solar Telescope Payload EGSE Platform Design and Development", Doctor Dissertation, National Astronomical Observatories, Chinese Academy of Sciences, Beijing, SC, 2009

[4]ZuoJiangtao, "RESEARCH OF REAL-TIME TELEMETRY DATA PROCESS BASED ON XTCE", Master Dissertation, Academy of Opto-Electronics, Chinese Academy of Sciences, Beijing, SC, 2011