Self-Configurable Flight Software Architecture with Goal-Oriented Operation for CubeSats

Cheng-Ting Wu¹, Ming-Yang Hong², Ya-Tzu Hung³, Jyh-Ching Juang⁴

Department of Electrical Engineering, National Cheng Kung University
No.1, Daxue Rd., 70101 TAINAN, TAIWAN - Tzu-Chiang Campus
Mail: ¹roger89632@gmail.com, ²hmie0117@gmail.com, ³e94011063@gmail.com, ⁴juang@mail.ncku.edu.tw

Abstract: To successfully perform a space mission, a good coordination of the ground segment and space segment is needed. Sending a sequences of time-based commands from ground segment to space segment is the most popular method for space mission operation. However, the time-based commands may not be efficient for CubeSat operation, especially when the mission becomes more complicated. In the paper, the use of goal-oriented operation is proposed and discussed as an alternative in ground segment operation. A self-configurable flight software architecture is proposed to facilitate goal-oriented operation for CubeSats. Under this architecture, the system parameters which are used to control the behavior of CubeSat are updated in an iterative manner so that the efficiency of the task decomposition task can be incrementally improved. Moreover, in this self-configurable architecture, a margin library is used to limit the system parameters in reasonable bounds to prevent the CubeSat from violating safety requirements. Thus, with goal-oriented operation and a well-designed self-configurable flight software architecture, the system can adapt to the unpredictable and harsh environment in space and also can progressively enhance the performance in operation.

1. INTRODUCTION

In recent year, CubeSats have been developed in different universities, research organizations, and industrial sectors around the world. Those low budget project with CubeSats are benefited from its short times for designing and launching into orbit. The well-designed flight software (FSW), which served as a brain of satellite, thus plays an important role in CubeSats. To successfully perform a space mission, it needs a good coordination between ground segment and space segment. Sending a sequences of time-based commands from ground segment to space segment is the most popular method for space mission operation. However, for CubeSat operation, the limitation of communication bandwidth and the short coverage duration in low-earth orbit (LEO), the time-based commands may not be efficient especially when the mission becomes more complicated. Hence, the advantage of strategy in goal-oriented operation is introduced reasonably. A self-configurable flight software architecture is proposed in this paper, aiming to facilitate the goal-oriented operation for CubeSats. With the proposed architecture, it has the potential to complete more difficult space mission while the cost of resources is reduced.

2. GOAL-ORIENTED OPERATION

The manner of sending time-based commands to CubeSat is able to achieve high confident by testing those commands previously. However, when the mission of satellite becomes more complicated, the issues such as receive fragmentary command during large time-based commands uplinked will appear. A desirable operation of goal-oriented, which is uplinked by high level goals instead of explicit commands, is introduced appositely. The concept of goal-oriented operation has been investigated in 1999, when
Remote Agent Experiment (RAX) first time demonstrates autonomous control of the Deep Space One spacecraft [1][2]. According to the manipulation of RAX, the conceptually use of goal-oriented operation is proposed and depicted in Fig. 1.

![Fig. 1 Goal-Oriented Operation](image)

In Fig. 1, the Goal Manager is responsible for dealing with high level goals sent from ground segment. It will verify whether a goal is invalid or not. If a goal cannot pass the verification, the signal for invalid goal will send back to ground segment through FSW.

While On-Board Planner receives a goal, it will decompose the high level goal into different subtasks in different hierarchy, and then decomposes each subtask into low level commands which can be executed by hardware. The On-Board Scheduler will dispatch and organize those low level command as a script, so that it can be subscribed by Scripted Executive and waiting for execution.

The function of Knowledge Base Model (KBM) is to list the equipment of its attribute and resource cost. This provides an important information for On-Board Planner during the decomposition procedure. Owing to the fact that there are alternative ways of decomposing a task, the parameters in KBM play as a significant linchpin.

The state in CubeSat will be identified by Fault Diagnosis as shown in Fig.2, this mechanism will also compare results of command with current state. If the comparison exists inconsistency, it will try to resent the command. Fault Diagnosis will recommend Reconfiguration System action reconfiguration while intolerable inconsistency still existed.

![Fig. 2 Goal-Oriented Operation](image)
With these modules mentioned above, the proposed use of goal-oriented operation can be performed in CubeSats. This increases the autonomous level of manipulation in space segment, hence, making more possible to achieve mission within more difficulty.

3. SELF-CONFIGURABLE DESIGN

After the satellite be launched, it’s hardly has chances to change the software inside unless going through the remote update process. However, even though the remote update may solve the bug found after the launch, it takes a long time to successfully transfer the replace code. Moreover, from the aspect of lifetime for those low budget project in CubeSat, it seems not a solution for using approach of remote update merely trying to enhance the performance in satellites. Thus, a well-designed self-configurable software architecture is proposed to adapt the rapid change in the harsh space environment and fulfill the requirements of mission at any time.

The FSW uses some parameters in module and system to refine or change the behavior of the software during the orbit. It may be necessary to tune or adjust those parameters depending on different situations. The use of self-configurable architecture has benefits to update parameters in an iterative manner so that the capability of CubeSats can be incrementally improved. A basic self-configurable mechanism is illustrated in Fig. 3.
In order to construct this architecture in FSW as shown in Fig. 3, several modules contributed to Self-Configurable Manager are needed, including margin library, value pool, update algorithm, and analyzing mechanism.

**Margin Library** is a general definition for each parameter, this library defines the upper and lower limit of the parameters. It will make sure that the updated value is reasonable and prevent the CubeSat within the safety requirements. It also serves as an indirectly function to eliminate the redundant way of decomposing tasks during action in Goal-Oriented Manager. Although the range of variable values will be restricted under this library, the defined range could be configured by the telecommand from ground station (GS).

**Value Pool** is a set of all values that the parameters used before. At first, only the initial value is in the pool. After several iterations, some new value will be produced into the pool. Those values in the pool have its rank according to performance of used parameters. This module helps the Update Algorithm to choose the most appropriate value.

**Update Algorithm** is a method to keep tracking on the values in value pool and update the parameters according to the calculation results from Analyzing Mechanism if necessary. In order to update the value, it must pass through the consideration of margin library and the outcome from analyzing mechanism. The module is illustrated as Fig. 4.

![Fig. 4 Illustration of Update Algorithm](image)

**Analyzing Mechanism** is a way to calculate and analyze the performance of FSW. A set of recording data will be generated to assess the effectiveness and sensitivity of a certain value, paving the way to determine the optimal combination of parameters. By running through this mechanism, the Update Algorithm is able to determine if the new value increase more performance or not. The mechanism of this module is depicted as Fig. 5.
With proposed components above, a self-configurable software architecture can be established. Under this architecture, it can adapt the unpredictable and harsh environment in space and can promote the performance of CubeSats. Furthermore, the proposed architecture can also facilitate goal-oriented operation by updating parameters used to control the operation, thus, find out an efficient decomposition and reducing the resource cost in CubeSats.

4. FUTURE TESTING SCENARIO

In the next step of testing, work will focus on verification of this architecture. The key point will be emphasized on power usage, available memory and as well as the duration of planning and scheduling. A well understanding of resources consumption can provide designer an important information about how much data budget margin it still has in this architecture. This information is beneficial for developer to look for the alternative payload in different criteria which still achieves the mission.

In order to test the robustness of this architecture perform in CubeSats, one of the testing scenario is to send a conflict command through debugging interface. Therefore, developer will figure out what actions have done by the software in order to resolve this anomaly, and also how much time the software take for reschedule and how many resources it will cost. The testing scenario is shown as Fig. 6.
5. CONCLUSIONS

Goal-oriented operation provides a mechanism for the efficient satellite operation and better utilization. If needed, there is a selection to switch into non-goal-oriented operation mode as the prevalent operational paradigm of command sequencing. For the proposed architecture, it’s still need to determine how to exploit the update algorithm with limited resource in CubeSats, and the implementation of proposed architecture also needs to be validated. However, the improvement of space technology and semiconductor industry make this architecture possible. Consequently, the proposed architecture would allow CubeSats to conduct more complicated missions.

6. ACKNOWLEDGMENT

The research was supported by Ministry of Science and Technology, Taiwan under grant MOST-106-2922-I-006 -091

7. REFERENCES