Scheduling for a Small Satellite for Remote Sensed Data Collection

Donovan Torgerson
Christoffer Korvalnd
Jeremy Straub
Ronald Marsh

Available at: https://works.bepress.com/jeremy_straub/149/
Abstract

Small satellites, such as CubeSats, serve as excellent platforms for the collection of data that can be supplied to a geographic information system. To serve this need, they require a robust and lightweight task scheduler due to their limited onboard power production capabilities as well as internal space restrictions. Because of these constraints, schedules must be optimized; however, the scheduling process must be performed using limited processing (CPU) power.

Several considerations must be taken into account in order to make a scheduler for these systems. This paper highlights requirements such as inter-dependency of onboard systems, and limited windows of communication with ground based operators. After considering the genetic algorithm, exhaustive search, and heuristic based approaches, it has been decided that the heuristic based approach is the most feasible for this application. Time and power constraints were highly scrutinized in this comparison in order to maximize performance. Using this scheduling system will also enable the efficient collection of remote sensed data, such as images.

Implementation

With the limited power, processing, and time available on a small satellite’s flight computer (shown in Figure 1), it is vital to maximize the efficiency of the onboard scheduling software. On top of this, inter-dependencies of onboard systems, shown in Figure 2, were also addressed. Thorough qualitative evaluation of genetic, exhaustive search, and heuristic algorithms resulted in the selection of a heuristic approach. Based on this approach, an operating software flowchart (shown in Figure 3) has been created.

This approach takes into consideration the priority of tasks, how much power and time are used for each, if the task is in within the earliest start date and latest completion date and how many tasks will be effected if the current task is chosen. Once all of these have been taken into consideration, each task is assigned a weight. The system then begins running tasks in order of weight, beginning with the task with the lowest weight.

Future Work

Future work includes completion of both the heuristic scheduling system as well as the OpenOrbiter operating software. Once complete, an assessment of the software based on NASA and Department of Defense Guidelines related to aerospace software systems and validation of compliance with Consultative Committee for Space Data Systems (CCSDS) communications standards (e.g., CCSDS 132.0-B-1 and CCSDS 133.0-B-1) will be performed.

Acknowledgements

Small satellite development work at the University of North Dakota is or has been supported by the North Dakota Space Grant Consortium, North Dakota NASA EPSCoR, the University of North Dakota Faculty Research Seed Money Committee, North Dakota EPSCoR (NSF Grant # EPS-814442), the Department of Computer Science, the John D. Odegard School of Aerospace Sciences and the National Aeronautics and Space Administration.

The involvement of the numerous students from multiple disciplines in this project is gratefully acknowledged. Also, thanks are given to the numerous faculty mentors who have helped make this project possible.

References