Paper ID: 34777
oral

SPACE OPERATIONS SYMPOSIUM (B6)
New Operations Concepts, Advanced Systems and Commercial Space Operations (2)

Author: Mr. Richard Duke
Surrey Space Centre - University of Surrey, United Kingdom, r.duke@surrey.ac.uk

Dr. Christopher P. Bridges
Surrey Space Centre, University of Surrey, United Kingdom, c.p.bridges@surrey.ac.uk
Mr. Brian Stewart
Surrey Space Centre - University of Surrey, United Kingdom, b.stewart@surrey.ac.uk
Dr. Ben Taylor
Surrey Space Centre, University of Surrey, United Kingdom, b.taylor@surrey.ac.uk
Ms. Chiara Massimiani
Surrey Space Centre, University of Surrey, United Kingdom, c.massimiani@surrey.ac.uk
Dr. Jason Forshaw
Surrey Space Centre, University of Surrey, United Kingdom, j.forshaw@surrey.ac.uk
Prof.Dr. Guglielmo Aglietti
Surrey Space Centre, University of Surrey, United Kingdom, g.aglietti@surrey.ac.uk

INTEGRATED FLIGHT & GROUND SOFTWARE FRAMEWORK FOR FAST MISSION TIMELINES

Abstract

Flight and ground segment software in university missions is often developed after hardware has matured sufficiently and also as bespoke codebases with very little common software to address key subsystems in power, communications, attitude, and payload control. This bespoke software process is often hardware specific, highly sequential, and costly in staff/monitory resources and, ultimately, development time. Within Surrey Space Centre (SSC), there are a number of satellite missions under development with similar delivery timelines that have overlapping requirements for the common tasks and additional payload handling. To address the needs of multiple missions with limited staff resources in a given delivery schedule, computing commonality for both flight and ground segment software is exploited by implementing a common set of flight tasks (or modules) which can be automatically generated into ground segment databases to deliver for advanced debugging support during system end-to-end test (SEET) and operations.

This paper focuses on the development, implementation, and testing of SSC's common software framework on the Stellenbosch ADCS stack and emulators for numerous missions including Alsat-1N, RemoveDebris, SME-SAT, and InflateSail. The framework uses a combination of open-source embedded and enterprise tools such as the FreeRTOS operating system coupled with rapid development templates used to auto-generate C and python codes offline from "message databases". In the flight software, a "core" packet router thread forwards messages to and from threads for inter process communication (IPC). On the ground, this is complemented with an auto-generated PostgreSQL database and web interface to test, log, and display results in our satellite operations centre. Profiling is performed using FreeRTOS primitives to manage module behaviour, context, time and memory – especially important during integration. This new framework has allowed for flight and ground software to be developed in parallel across SSC's current and future missions faster, with fewer propagated errors, and increased consistency between the flight software, ground station and project documentation.