IAF SPACE SYSTEMS SYMPOSIUM (D1)
Interactive Presentations - IAF SPACE SYSTEMS SYMPOSIUM (IP)

Author: Mr. Pierre-Baptiste LAMBERT
Centre National d'Etudes Spatiales (CNES), France

Mr. Julien Galizzi
Centre National d'Etudes Spatiales (CNES), France
Mr. Thomas DELMAS
Centre National d'Etudes Spatiales (CNES), France
Mr. Simon Corbin
Centre National d'Etudes Spatiales (CNES), France

FASTER, BETTER, CHEAPER: THE FLIGHT SOFTWARE ANSWER

Abstract

Based on the Integrated Modular Avionics paradigm from the aeronautics field, which intends at hosting various segregated software applications of various criticality levels on a same on-board computer (OBC), the French space agency (CNES) developed, with a similar approach, its Flight SoftWare (FSW) framework: **LVCUGEN** (name subject to change).

LVCUGEN is based on the Time-and-Space Partitioning (TSP) architecture, making use of a hypervisor (typically XNG from Fentiss) to orchestrate software partitions, each of them hosting a specific application (IO piloting, Monitoring&Control with PUS protocol, events management, memory management, mission functions, . . .). Each partition operates as a virtual machine would, with its allocated CPU time slots and memory space, not shared with any other partition. Developed over the last decade, LVCUGEN addresses the following challenges:

- **Multi-provider software development:** each partition being segregated from the others, it can be tested and validated on its own.
- **Multiple criticality levels in the same FSW:** the same OBC can host lower quality assurance level applications (typically mission software) with higher quality assurance level applications (HW management, . . .).
- **Reduced time to market through component reuse:** each partition can be considered as a modular component, either to be used "as-is" or instantiated (in case of partition devkits), almost hardware agnostic. Recurring partitions are already qualified ECSS level B with a DevOps process in place for continuous integration.
- **At system level, reduced mass and power requirement:** hosting various software functions in the same OBC avoids the need for multiple OBCs (one for platform, one for GNSS, one for STR, one for mass memory, etc.) which would each use a very small percentage of their capabilities.

The article presents LVCUGEN and shows that it is a relevant baseline for today's platform, payload or equipment FSW for all kinds of missions: institutional missions (e.g. CNES-DLR-JAXA MMX rover, mass memory FSW for SWOT or MERLIN), new space missions (e.g. N3SS from U-SPACE, KINEIS constellation from HEMERIA), scientific instruments (e.g. MAJIS aboard JUICE mission, ECLAIRs & MXT aboard SVOM mission, . . .), student nanosatellites (e.g. Nanolab academy Eyesat, Aerosat, ...), etc.

The article also describes the roadmap for LVCUGEN, noticeably facilitating access by going open source (pending reflection), offering a Linux-based “app store” partition for easy application embeddability, adapting to swarm missions with new autonomy and network features, applying MBSE process to this kind of architecture, etc.