A Re-usable Spacecraft Flight Software Architecture

cFS is both platform and project independent

Description
The Core Flight System (cFS) is a platform and project independent reusable software framework and set of reusable software applications. There are three key aspects to the cFS architecture: a dynamic run-time environment, layered software, and a component based design. It is the combination of these key aspects along with an implementation targeted to the embedded software domain developed using NASA Procedural Requirements (NPR) that makes the cFS suitable for reuse on any number of NASA flight projects and/or embedded software systems at very significant cost savings.

The architecture layers include a RTOS/BOOT layer, an Abstraction Library layer including an Operating System Abstraction Library (OSAL) and Platform Support Package (PSP) library, core Flight Executive (cFE) layer made up of five core API services, and an Application layer containing a set of re-usable libraries and Command and Data Handling (C&DH) applications. The Abstraction library layer runs on top of the RTOS/BOOT layer. The cFE is layered on top of the OSAL and PSP. And the applications are layered on top of the cFE. Each component in the system is a separate loadable file.

The cFS flight software framework takes advantage of a rich heritage of successful Goddard Space Flight Center flight software efforts and addresses the challenges of rapidly increasing software development costs and schedules due to the constant changes and advancements in hardware. Flight software complexity is expected to cultivate dramatically in coming years and the cFS provides a means to manage the growth and accommodate changes in flight system designs.

To support reuse and project independence, the architecture contains a configurable set of requirements and code. The configurable parameters allow the cFS to be tailored for each environment including desk-top and closed loop simulation environments. The ability to run and test software applications on a developer's desktop and then deploy that same software without changes to the embedded system is now possible using the cFS. Science and mission software can be developed and functionally tested very early in the project and well before any project hardware is even available. The architecture contains a tool suite which includes a reusable test suite. In addition, the architecture contains reusable artifacts including requirements, design documentation, test procedures and results, development standards, and user guides. The cFS architecture simplifies the flight software development process by providing the underlying infrastructure and hosting a runtime environment for development of project/mision specific applications. This cFS architecture also simplifies the flight software maintenance process by providing the ability to change software components during development or in flight without having to restart or reboot the system.

Benefits
- Reduces time to deploy high quality flight software
- Reduces project schedule and cost uncertainty
- Software reuse is facilitated and formalized
- Enables collaboration across organizations
- Simplifies flight software sustaining engineering
- Advanced concepts and prototyping platform

Applications
The cFS architecture is suited for use in any embedded software command and control computing system. The cFS is currently in use on several missions and projects. Some current users include:
- GSFC Lunar Reconnaissance Orbiter mission
- APL Radiation Belt Storm Probes mission
- JSC Morphewas prototype, 1.6B and 1.5C vehicles
- ARC Lunar Atmosphere and Dust Environment Explorer
- GSFC/JAXA Global Precipitation Measurement mission
- GSFC Magnetosphere Multi-scale mission
- HH/PLS Solar Probe Plus
- Emergent Space Technologies, Incorporated, Cluster Flight Algorithms and Flight Software project
- GSFC ISEM Spacecube
- Moon Express, Incorporated, Google Lunar XPRIZE robotic lunar spacecraft
- Odyssey Space Research, LLC, NEySys Project