Command and Execute

Commander is a flight software suite for plan execution in a fleet of one or more vehicles. Commander uses a hierarchical execution model that divides control between a single centralized coordinator and distributed executors on every vehicle. All software is designed for execution onboard flight vehicles. Commander interacts with other Emergent flight software products to enable advanced capabilities on spacecraft fleets, such as: coordinated orbital maneuvers and relative navigation, fault detection, isolation and recovery, distributed computing, and simulation and visualization.

Commander satisfies a need for reconfigurable management of subsystem applications by executing a plan. The plan specifies telemetry (incoming messages) and commands (outgoing messages) that should be processed to control subsystem applications. Customization of the execution plan enables human-in-the-loop, human-on-the-loop, or fully autonomous operational modes.

Commander: Reconfigurable Plan Execution

Mission 1: Noncooperative inspection mission

Commander configures Autopilot and sends periodic commands to enter new orbits. Commander monitors responses to confirm commands are accepted. Ground operator approval is required before each subsequent orbit change.

Mission 2: Earth observing scheduling

Commander sends attitude commands according to a precomputed schedule. Commander monitors telemetry from Guardian to enter a safe mode while recovery is attempted.
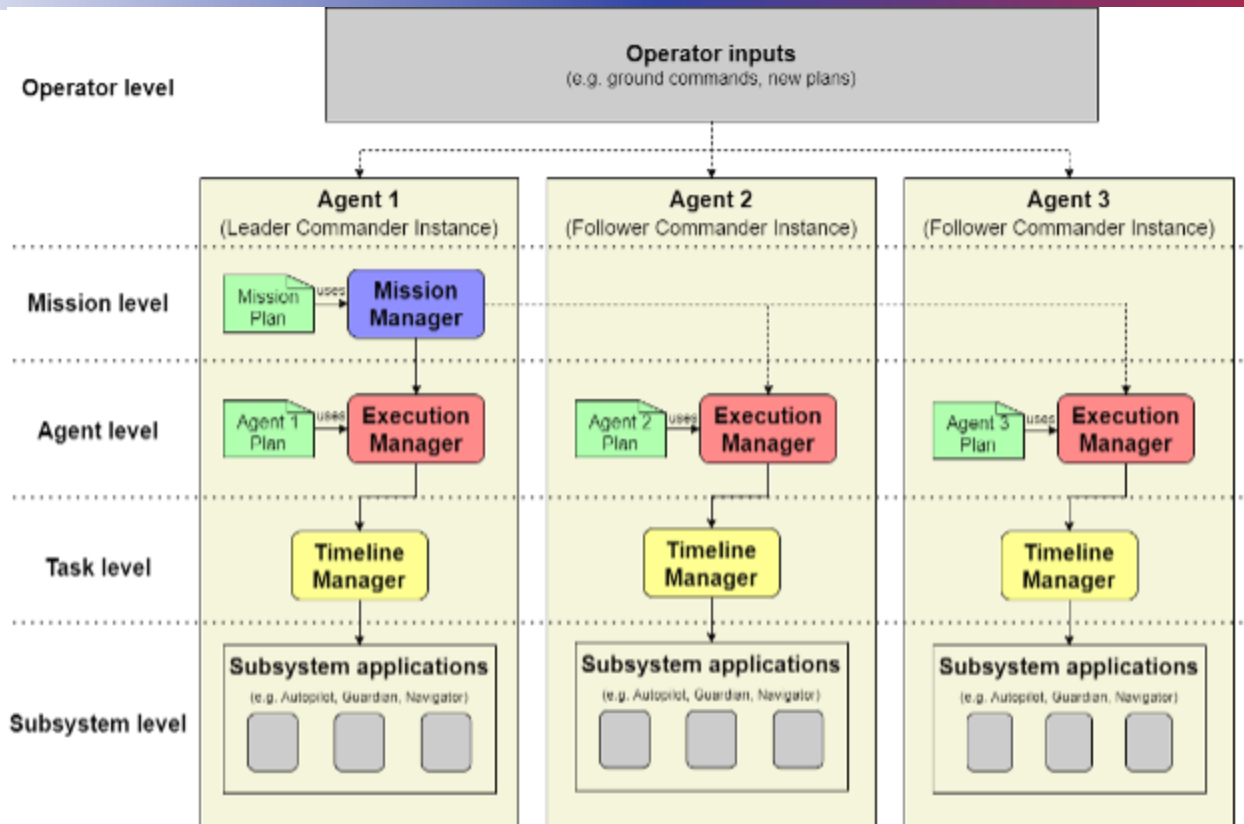
Modules/Components

Mission Manager

Coordinates fleet activities and responds to ground commands using an event-driven finite state machine. Responds to telemetry from Execution Manager(s) to manage agent-level applications.

Execution Manager

Coordinates activities on a single vehicle using an event-driven finite state machine. Queues tasks in Timeline Manager in response to received commands or application telemetry. Configures and responds to telemetry from subsystem-level applications.

Timeline Manager

Maintains a timeline of tasks. Performs task queueing and deconfliction based on priority and resource needs. Sends task commands to subsystem-level applications at appropriate times.

Specifications

- Linux-based OS
- Runs on x86 and ARM processors
- Configures managed applications
- Receives ground commands to change behaviors or adopt new plans
- Sends commands ro applications in response to received telemetry
- Developed in Docker running Ubuntu 18.04 LTS
- Uses Emergent's Gear framework for messaging and middleware compatibility
- Executes a plan to manage subsystem applications to enable long-term autonomous and semi-autonomous operations

About Emergent

Emergent Space Technologies, Inc. researches, develops, integrates, and tests advanced systems and software solutions for civil, military and commercial space missions. We are industry leaders in the development of flight software for multi-spacecraft missions, including constellations, formations and clusters of small satellites. Our core competencies are systems engineering, integration and test; guidance, navigation and control; orbital mechanics; positioning, navigation and timing; advanced modeling and simulation; and SW architecture, design, development and test. Our domain expertise and experience, combined with our knowledge of current and emerging technology, make Emergent the team of choice in the aerospace industry.