JOHNS HOPKINS APPLIED PHYSICS LABORATORY

ADVANCED MULTI-MISSION OPERATIONS SYSTEM

MIME: Mission Independent Memory Examiner

Ground Software for Tracking & Examining Spacecraft Memory

Overview

Many space missions have the need to:

- Track spacecraft flight software parameters, and other objects in memory, for automated & manual verification, analysis, and anomaly investigation
- Compare predicted versus actual configurations of spacecraft memory
- Verify and synchronize configurations against and between spacecraft and simulator(s)

MIME is ground software which delivers these memory object management capabilities by monitoring and interpreting spacecraft commanding and telemetry. MIME collects evidence (e.g. uplinked commands, downlinked command execution history, downlinked memory contents or CRCs, et al.) and sends them to mission-provided plugins to calculate predicted or measured impacts on spacecraft memory. MIME then records and leverages these predictions and measurements to give its users visibility into:

- All historical states of a single memory location over time
- The current or past state, at any given instant, of groups of memory locations
- Differences in memory state between two spacecraft or simulators at the current or a past instant
- Differences in memory state between the expected and actual states of memory locations
- Memory locations whose measurements do not correspond to any related predictions

MIME achieves its mission independence by making no assumptions about spacecraft, their memory object organization, or their flight software behavior. Missions adapt MIME by providing:

- One or many Mission-Specific Plugins, which emulate portions of spacecraft flight software behavior
- Configuration containing descriptions of all mission spacecraft and simulators and their memory organization, types, and flight software versions

MIME integrates natively with the AMMOS Mission Data Processing and Control System (AMPCS.) Additional integrations with other commanding & telemetry suites are planned.

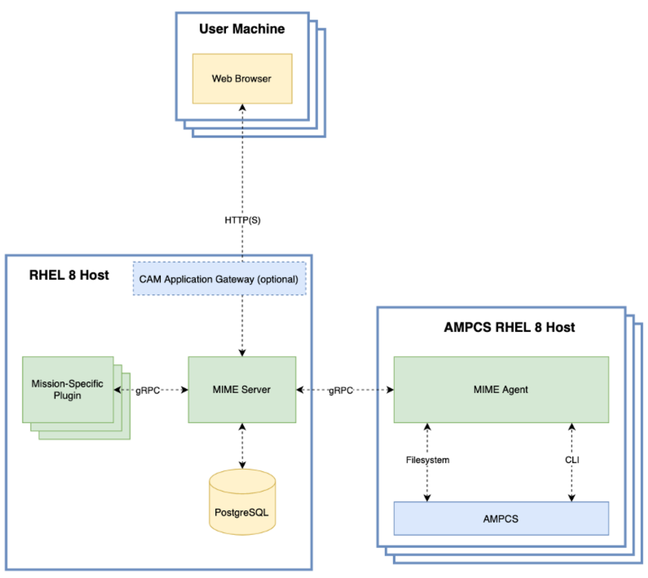
MIME is expected to be run primarily in a mission operations center (MOC), as well as in development, integration, and testing environments.

Missions which leverage MIME to implement their memory object management requirements may be able to reduce their operational cost and risk by reusing the mission independent components of MIME and by limiting software development efforts to the plugin(s).

Architecture & Design

MIME is architected and distributed as a set of application components:

1. **MIME Server**, an instance of which implements all primary functions, including memory object tracking and evaluation
2. **MIME Agent**, instances of which are responsible for collecting commanding and telemetry data
3. **Mission-Specific Plugins,** instances of which are responsible for emulating mission-specific flight software behavior as it relates to commanding and telemetry involving memory objects



Notional MIME Deployment Architecture

MIME deployments are composable: the number and location (host machine) of components running on any given mission ground data system (GDS) may be decided by the mission according to its requirements and nature. The composable aspects of the deployed system include:

- The location of a single MIME Server instance
- The location of a single PostgreSQL database
- The number and locations of any number of MIME Agent instances
- The number and locations of any number of Mission-Specific Plugin instances

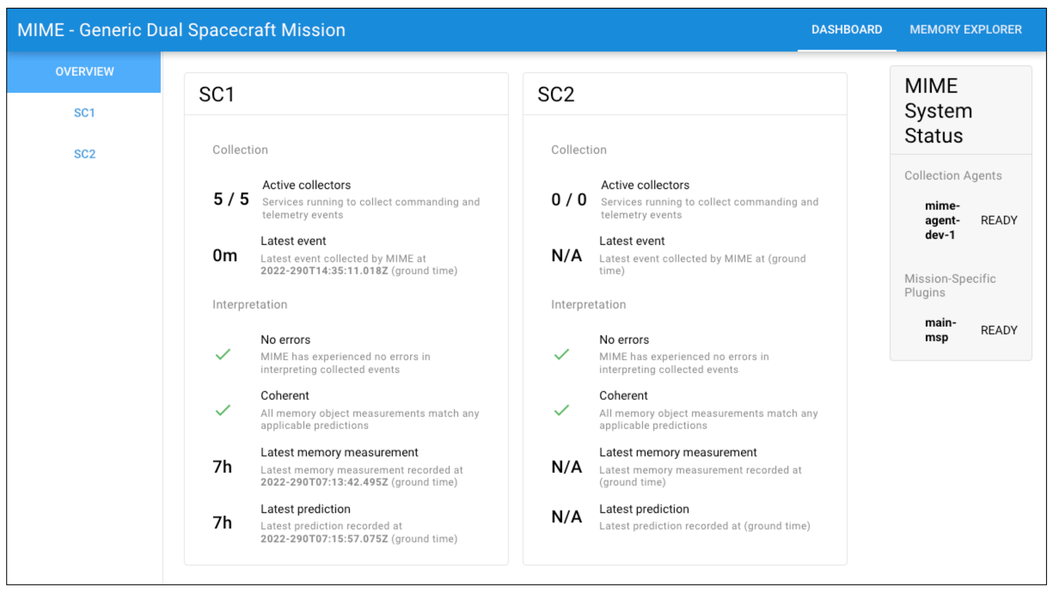
Additional design aspects make MIME well-suited for many missions, including aspects that were put in place based on the lessons learned from prior memory object management tools, software, and practices, including:

- Interpreting commanding & telemetry events that arrive out-of-order
- Graceful, visible, and recoverable degradation of memory tracking under most situations, including errors in interpreting commanding & telemetry events
- Support for multiple flight software versions per spacecraft and heterogenous memory organizations
- Support for bulk memory changes (e.g. spacecraft reboot, flight software updates)
- Support for automated I&T and related activities via its REST API
- Offering data export into common formats
- Support for autonomy & spacecraft-initiated changes (i.e. changes that are not associated with a collected piece of commanding or telemetry evidence)

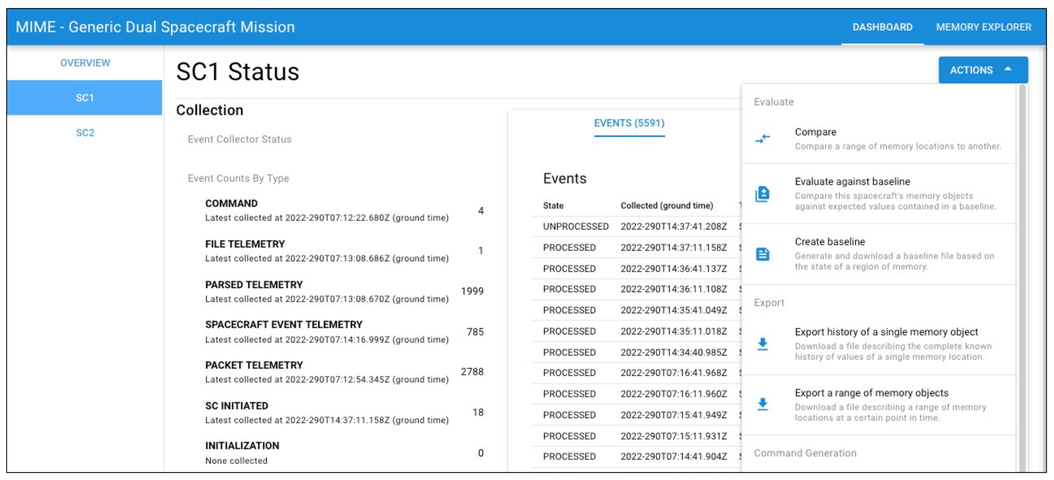
Points of Contact

MIME: robert.seng@jhuapl.edu
AMMOS: https://ammos.nasa.gov/contact

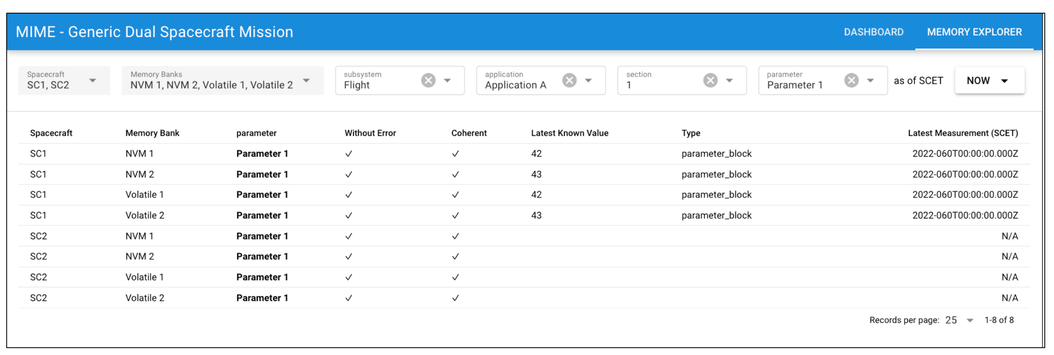
User Interface



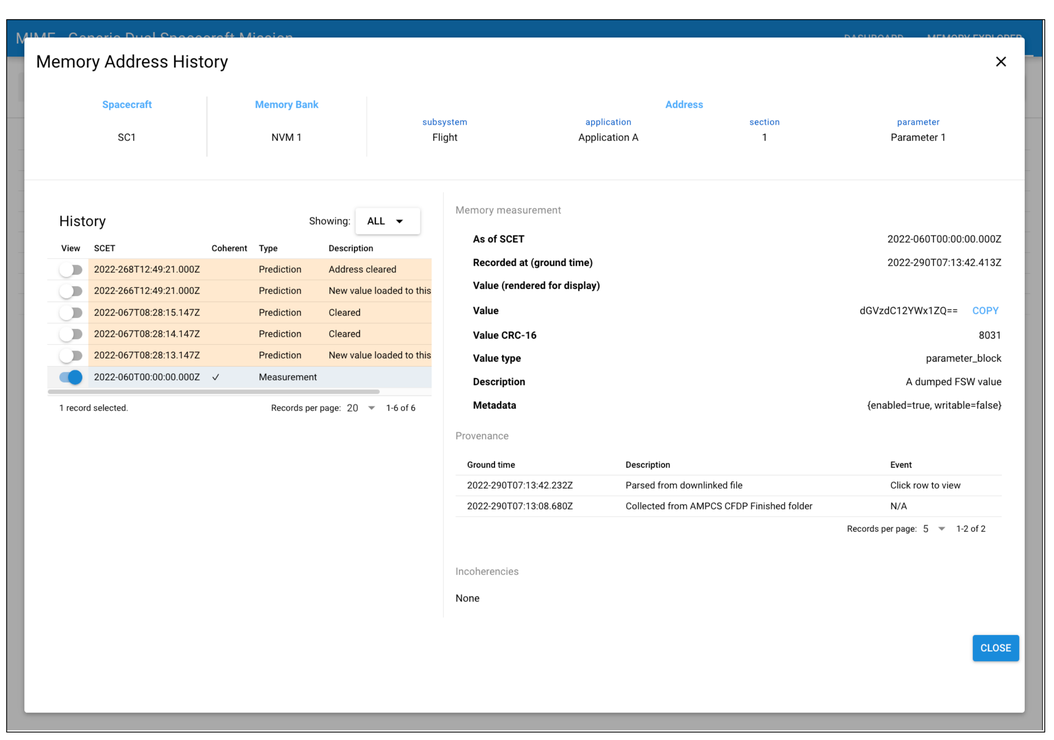
The Dashboard Overview shows all spacecraft and simulators the system is tracking, along with their commanding & telemetry event collection status, their event interpretation status, and their overall memory coherency (whether all memory measurements are congruous with all predictions.)



The Spacecraft Status view shows both a summarized view of spacecraft event collection and interpretation statistics, as well as a table that allows users to drill down into individual events and see their details. The Spacecraft Actions menu contains the entry point to all comparison, export, command generation, and maintenance functions.



The Memory Explorer view enables users to navigate the hierarchical memory organization of all tracked spacecraft. Users can view summarized metadata about the coherency and recency of tracked memory at any level, down to individual memory locations. It shows the latest data by default. It also allows users to view the tracking information as it was at any historical point in time.



The Memory Address History view shows users the complete known history of any single memory location, including all related memory measurements and predictions. Each measurement or prediction contains a time-ordered record of its provenance.