Trusted Autonomy and the NASA Platform for Autonomous Systems (NPAS)



Presentation to the Space Trusted Autonomy Group

All the tools required to implement hierarchical, distributed, safety critical, autonomous operations in an integrated software platform that is affordable, reusable, and scalable



NASA SSC Autonomous Systems Laboratory Team
Fernando Figueroa
May 19, 2022

NPAS Team

NASA SSC

- Project Manager: Dr. Lauren Underwood
- Project PI and Technical Manager: Dr. Fernando Figueroa
- Development Engineer: Landon Tynes
- Development Engineer: Brendan Cheng

D2K Technologies

- Autonomous Systems SME: Mark Walker
- Product Owner: Jon Morris
- Senior Software Engineer: Joshua Broberg
- Senior Software Engineer: Quentin Oswald
- Software Engineer: Brian Rey
- Scrum Master / Agile Coach: Michael Walker

Why NPAS

Why is NPAS important?

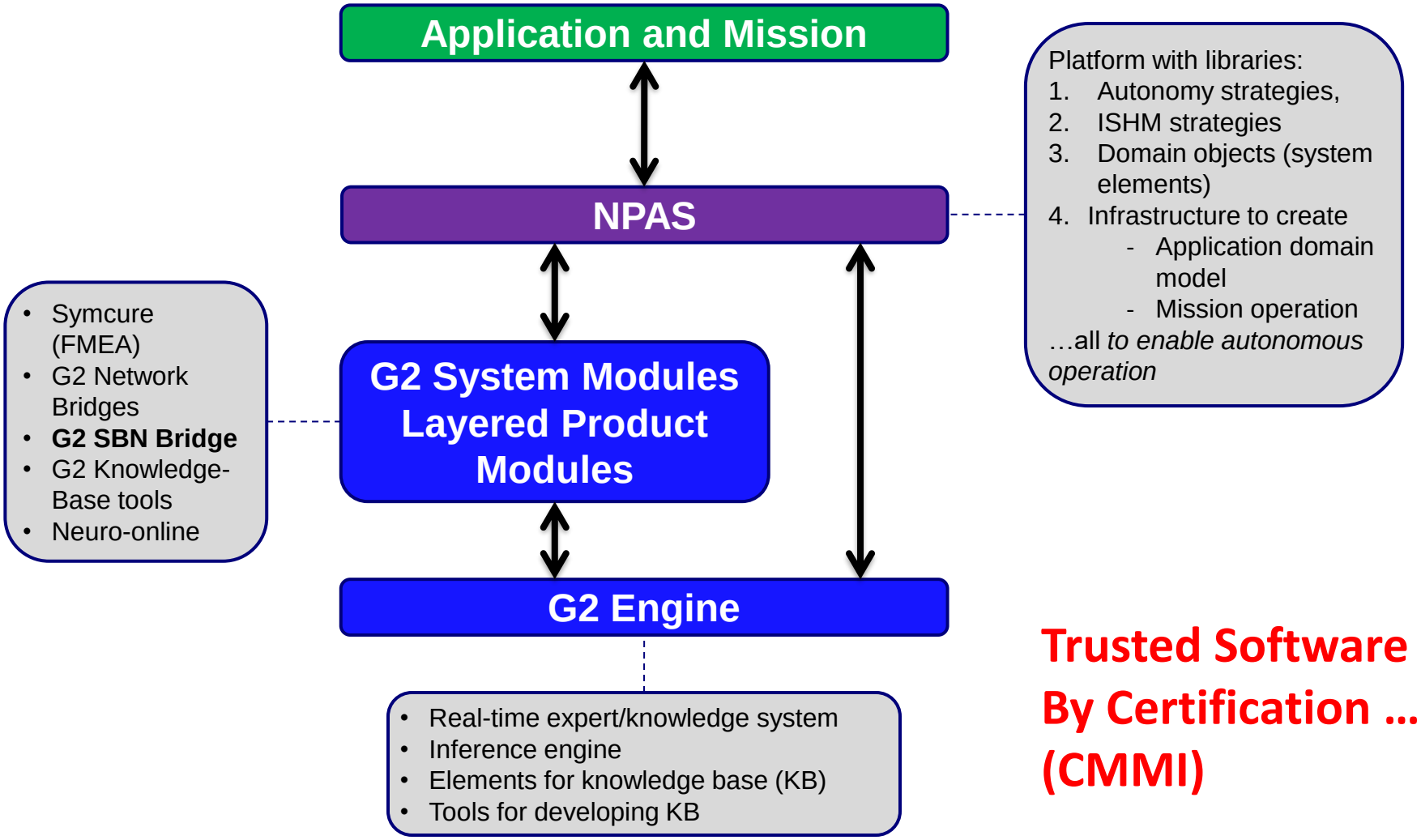
Artemis autonomous systems require

- ✓ Ability to operate without control from the crew or MCC
- ✓ Independent reasoning so persistent human inputs are not required
- ✓ Intrinsic monitoring of tasks, systems, components, etc.
- ✓ Rapid notification to crew/operators when necessary
- ✓ Visibility into decision processes – no black box algorithms

Current NPAS Objectives

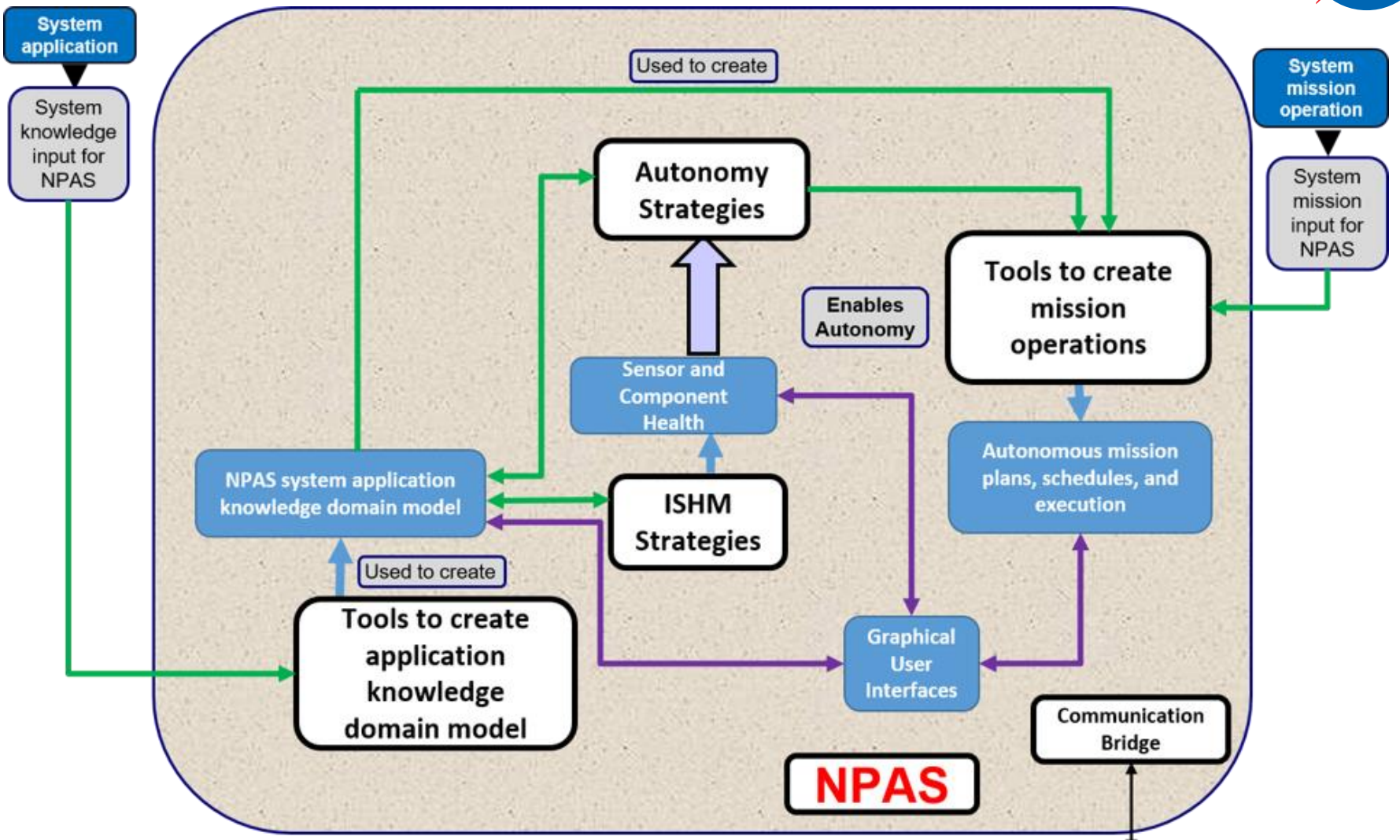
- Continue to reduce the time and cost to design, build, test and deploy robust, intelligent, distributed, hierarchical autonomous systems.
- Meeting NASA Gateway autonomy requirements and con-ops
- Implementing Gateway avionics hardware architecture (network protocols and message definitions)
- Class A certification of NPAS

Architecture for NPAS Implementation



Trusted Software By Certification … (CMMI)

Functional NPAS Architecture



Trusted Integration of all Autonomy functions

What is NPAS

NPAS is a platform built with G2

- NPAS technology development extends the inherent capabilities of G2
- NPAS encompasses the tools to simplify the creation of robust knowledge-based autonomous systems
- NPAS is currently in operational use at NASA SSC's High Pressure Gas Facility, a critical piece of test infrastructure used to support SLS, DoD, and commercial propulsion testing

NPAS provides tools for creating autonomous systems:

1. **ISHM strategies** for integrated system health management, including diagnosis, prognosis, and FMEA
2. **Autonomy strategies** framework based on concepts of operation, redundancy, nominal and off-nominal operation
3. **Domain object libraries** –reusable software, thus cost effective
4. **Infrastructure** to create **application domain models** and **mission operations that are scalable**

✓ **G2 provides foundational tools**
✓ **NPAS builds upon G2 capabilities to enable autonomous systems**

G2 is an expert system framework spun out from MIT

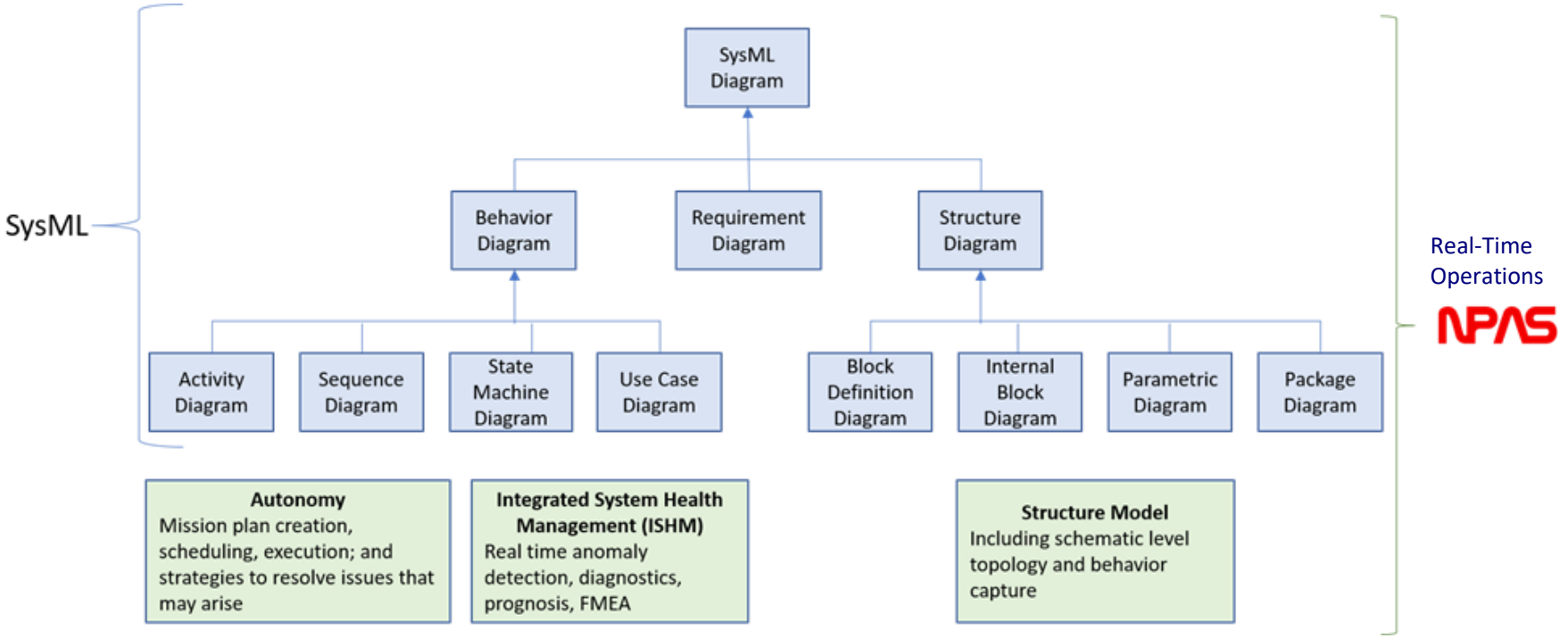
- G2 is a COTS tool for creating intelligent applications
- G2 is used in industry and DoD to control critical systems
- NASA uses G2 to train NASA astronauts to operate ISS payloads (JSC/MSFC)

NPAS is an integrated single platform for autonomy with cost-effective, reusable infrastructure and provides comprehensive system awareness to operators and users

Model-Based Systems Engineering, Real-Time Operations and Autonomy

G2/NPAS Model Development

- Includes content associated with all SysML diagrams
- Augments behavior content for ISHM and Autonomy
- Augments detail of structure diagrams by including schematic details
- NPAS uses model in real-time autonomous operations



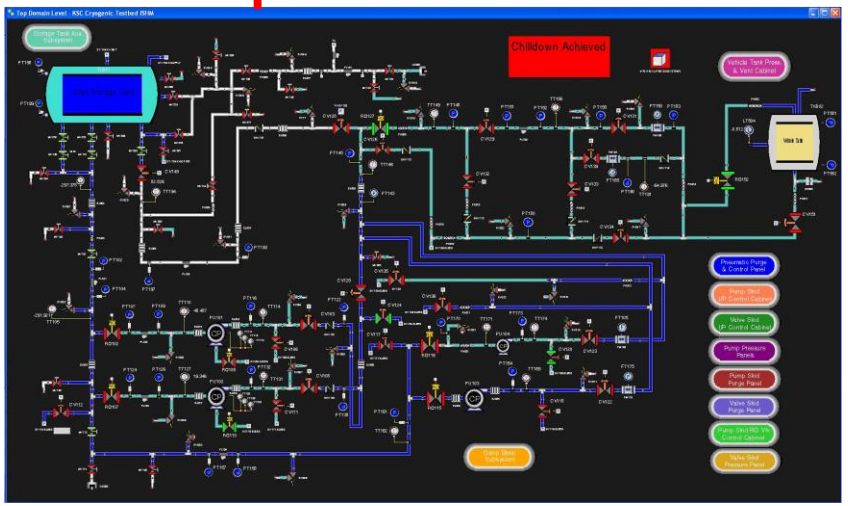
Trusted and complete reference domain description

Digital Twins: NASA Platform for Autonomous Systems (NPAS)

Stennis Space Center

Trust embodied in use of DT of the system for operations

- NASA Stennis Space Center has developed and continues to evolve the NASA Platform for Autonomous Systems (NPAS) (https://techport.nasa.gov/view/94884)
- NPAS-based autonomous operations utilize a software domain reference model—a ”digital twin” (DT) for mission management as well as for detection, diagnosis/prognosis, and resolution of anomalies
- NPAS supports the creation of such domain specific DTs using built-in and extensible component model libraries. These model libraries are typically designed to match the level of representation included in provided schematics and Interface Control Documents (ICDs)
- One benefit of the digital twin model is the provision of a dynamic system state model that the autonomous software can reference in real-time. This enables the software to make use of both topological and operational context when employing its reasoning logic
- NPAS Digital Twin domain representations are
  - ***ontological*** (e.g., they are composed of model constructs that map well to the ontologies present within the systems, processes, and equipment)
  - ***topological*** (e.g., they include the components, the interconnections, and the associations between the components in an operational context), and
  - ***dynamic*** (e.g., they are populated with live telemetry data and real-time state assessment regarding health and operational availability).
  - The representations are incorporated into the autonomous system manager software architecture seamlessly so that regardless of domain, all applicable measurement data can be properly integrated into a state model.



Digital Twin of a fluid transfer system



SSC Autonomous Systems Lab Autonomous Space Habitat Mockup

NPAS - Infrastructure

Stennis Space Center

Initial Domain Object Library and Knowledge

fluid



electrical



networks



mechanical



Trust from reusable libraries and infrastructure

Health Management



On board reasoning and decision making, using physics models, functional relationships, constraints, and concepts of operation



FMEA/diagnosis, prognosis, anomaly detection



Processes and event management





OODA loop

Autonomy

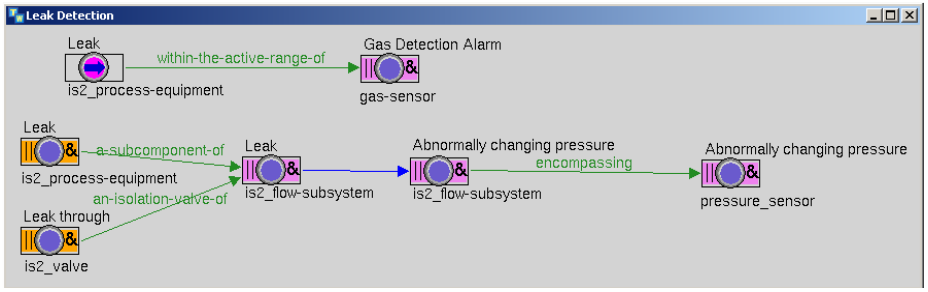
Plan, creation, scheduling & execution

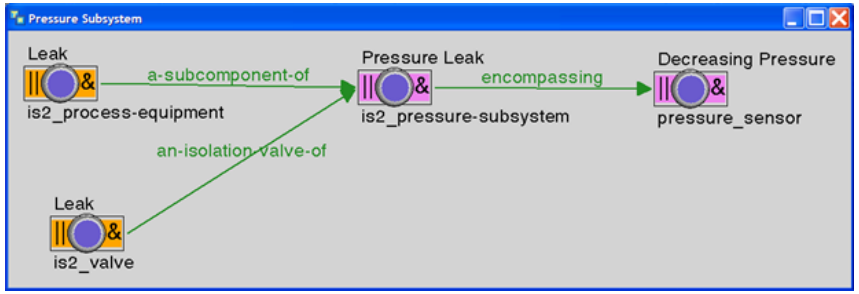
Foundational NPAS Infrastructure

NPAS Failure Modes and Effects Analysis (FMEA)

MIL-STD-1629A (NOTICE 2), Military Standard: Procedures for performing a Failure Mode, Effects, and criticality analysis (28 NOV 1984)

ID #	Item-Functional Identification	Function	Failure Modes and Causes	Mission Phase-Operational Mode	Failure Effects		Failure Detection Method
					Local End Effects	Next Higher Level	
	Process Equipment	Fluid feed subsystem	Leak	Sealed subsystem maintaining pressure		Pressure leak	Decreasing pressure measurement	Identify sealed subsystem, and check pressure sensors for decreasing pressure.

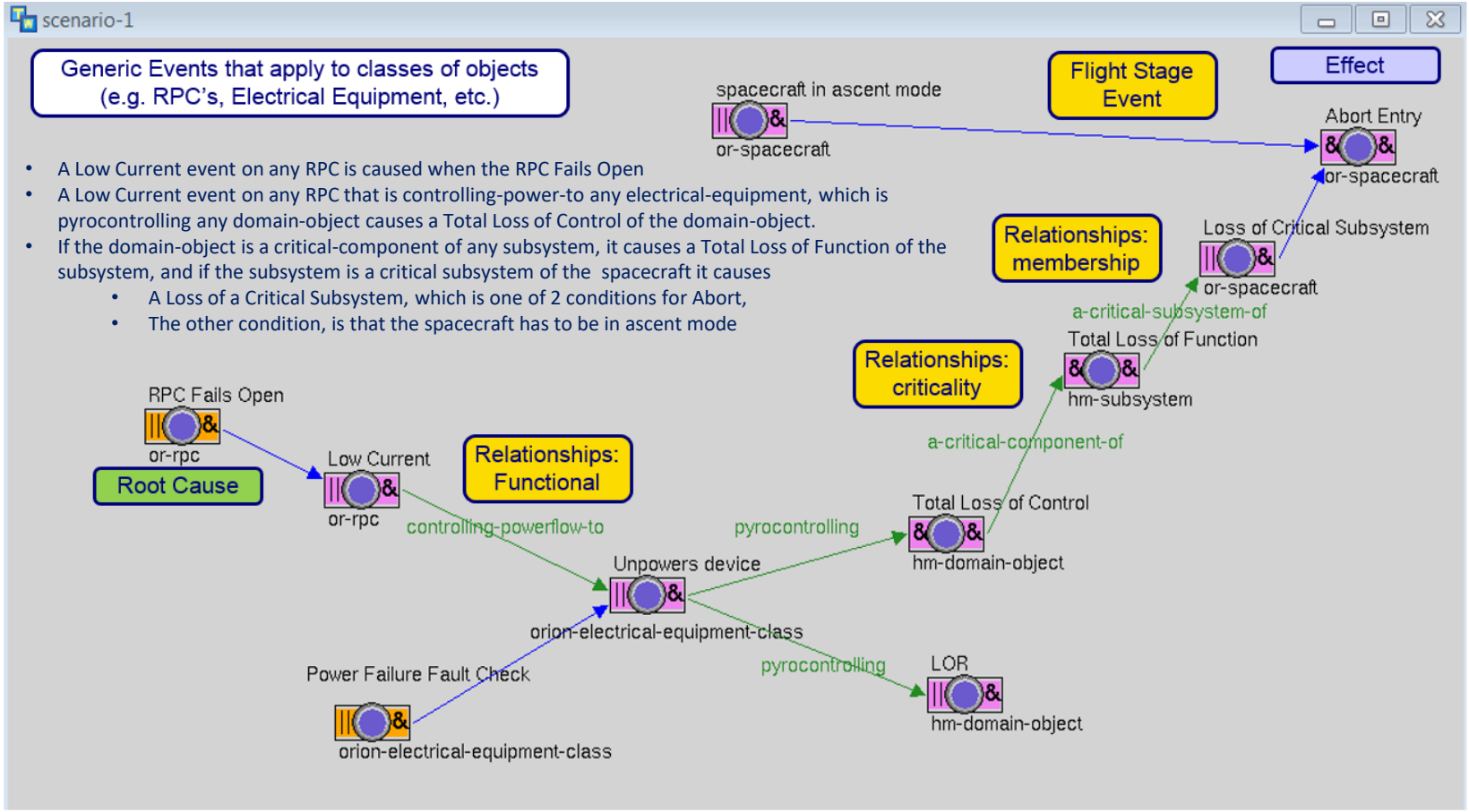




Two rich examples of implementation of leak detection root cause trees and encompasses Mil Standard and how FMEAs are defined. Right: leak event and consequences of leak; left: same leak event used for another diagnosis.
Root cause trees re-usable/generic, events can be used in multiple root cause trees

Trust from code that captures FMEA in structured natural language

Root Cause Leading to Abort

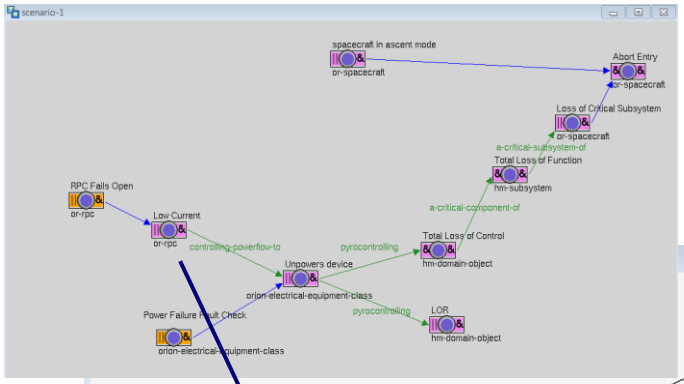


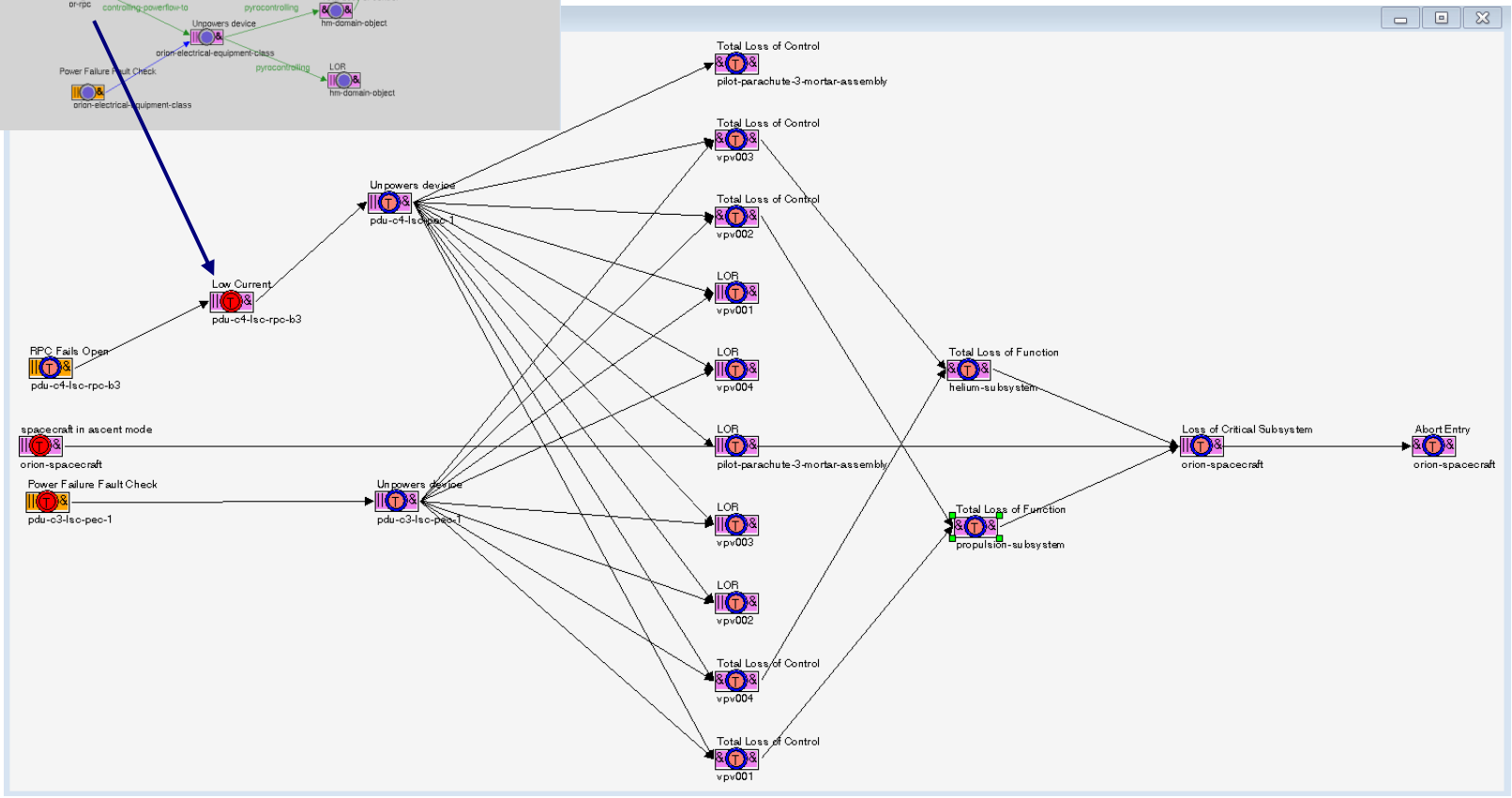
The system can send an event to any instance represented by a node. Causes and consequences will propagate upstream and downstream to assert status of all events (true, false, suspect).

Sample Specific Fault Model automatically generated when a particular RPC fails open

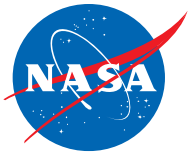
When RPC (PDU-c4-lsc-rpc-b3) experiences a low current event

- **the full blown specific root cause tree is generated**





NPAS is used to implement "Thinking Autonomy"



Trust of comprehensive coverage through on-board "thinking" encompassing intelligence, analysis, and decision-making

Brute Force **Minimal or no analysis and decisions by the system**	**"Thinking" Autonomy-** **Analysis and decisions are made by the system**
Problems solved by the expert off-line, and instantiated into computer code and **then the system chooses solutions corresponding to each prescribed case (lookup tables)**	**Problems solved by the autonomous system, on board in real time.** Analysis, reasoning, and decisions are evoked on-board by the system to derive information, knowledge, and optimal solutions
Solutions must be found for **every possible case defined for every object, every process that involves the object, and every configuration of the system where the object belongs (hard-coded)**; **not viable for the sustained evolution of autonomy**	The **autonomous system analyzes the system behavior at a high level of abstraction, by applying generic models** (e.g. physics models) that employ definitions and concepts and systematic processes; solutions are determined with a relatively small set of models
Coverage is not comprehensive; it is limited by the thoroughness of the analysis by expert off-line programmers	**Coverage is comprehensive** for each generic model used; total comprehensive coverage is feasible, since first principles cover all behaviors
The autonomous system does not "think" it merely makes use of what has been "thought" by humans	**The autonomous system "thinks"** to provide solutions that need not be solved off-line by humans

NPAS Implementations

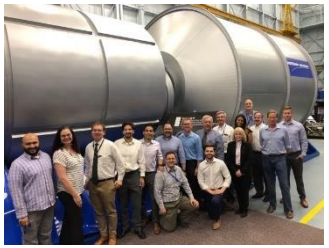
Stennis Space Center

NextSTEP-2 A: Habitation Systems

Northrop Grumman Innovative Systems (NGIS)

Developed prototype hierarchical distributed autonomous operation capability, including an **NPAS Vehicle Manager** and NPAS **Electrical Power System**

- ✓ Integrated in Habitation in the Life (HITL) demo
- ✓ Included in NGIS documentation deliverables
- ✓ Enhanced NPAS Vehicle Manager UI created and presented at NGIS's SME stand alone demonstration



Developed a baseline hierarchical distributed autonomous operation capability

NPAS Operational on Flight Processor

SSC Cooperative Agreement Contract with Ignite Technologies

Demonstrated NPAS/G2 functionality on a space qualified processor

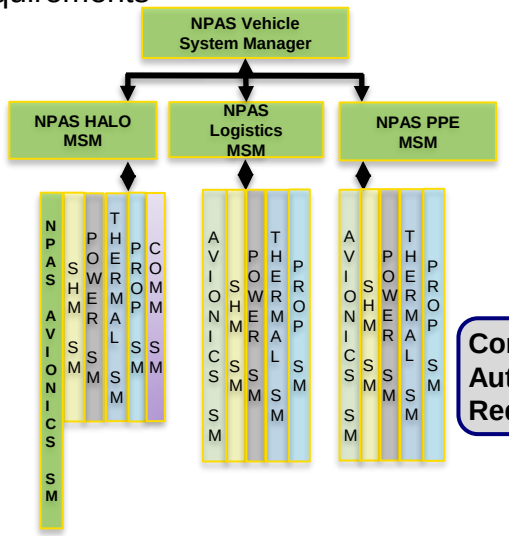




AiTech SP0-S/RAD 750
3U cPCI Radiation Tolerant Power PC SBC

Gateway Autonomy Architecture Demo

Demonstrated hierarchical distributed autonomy architecture prototype aligned with Gateway Autonomy ConOps, VSM Concept of Operations and Gateway VSM requirements



Conforms to Gateway Autonomy Requirements

Demonstrate G2 SBN Bridge

- ✓ **Demonstrate G2-Software Bus Network (SBN) bridge for integration with cFS applications**
- ✓ Overall performance testing - bridge functionality behaved as expected

Enabled NPAS to communicate with cFS app

AES Technology Infusion – Northrop Grumman, NextSTEP-2

NextSTEP-2, Habitation – NGIS Integration

- ✓ Quickly created prototype NPAS Vehicle Manager and NPAS EPS (Electrical Power System), that included hierarchical distributed autonomy
- ✓ Integrated with existing NGIS Station Telemetry Interface
- ✓ Completed in abbreviated time – during government shutdown
- ✓ Able to generate additional UIs in response to customer request
- ✓ All deliverables met within 5 months
- ✓ Positive feedback from HITL astronaut

Northrop Grumman Awarded NASA Contract to Provide First Crew Module for Artemis Program Gateway



NPAS VM User Interface

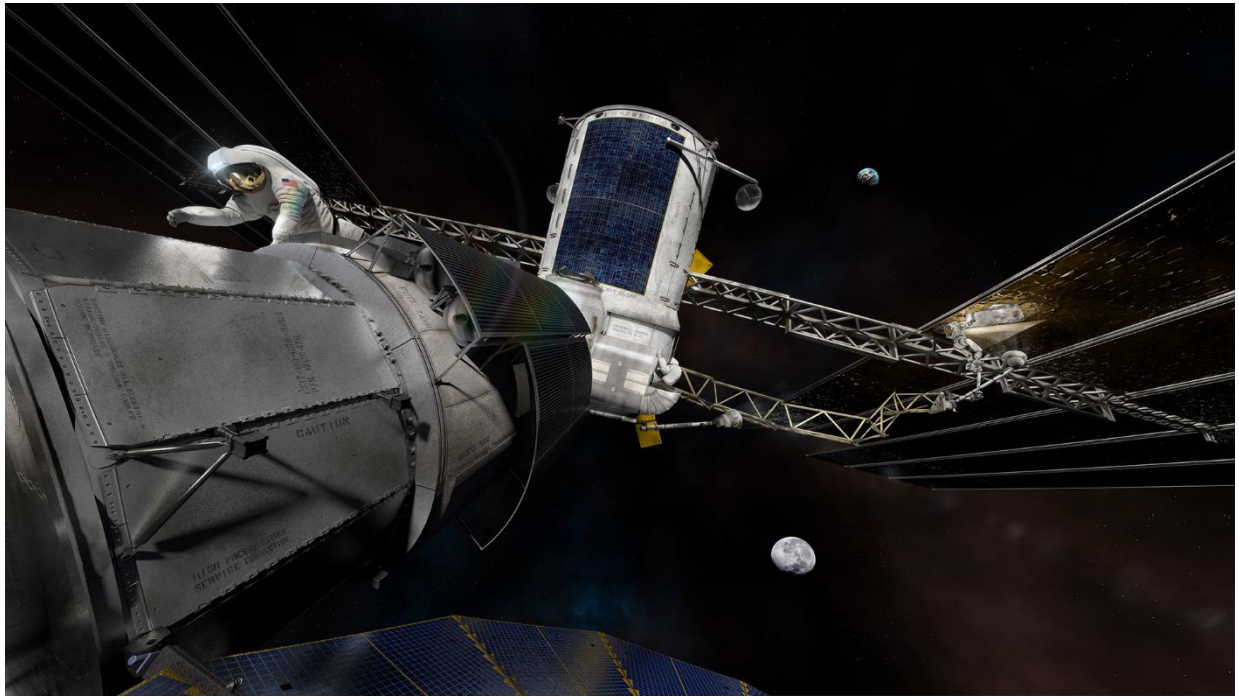
Trust from standardized network integration

AES Technology Infusion FY20

Autonomous Avionics Risk Reduction with Lockheed Martin

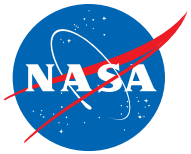
Objectives:

- Autonomous avionics activity to help reduce risk for demonstrating autonomous operations between Gateway and Orion
- Supports efforts to demonstrate a distributed hierarchical autonomy implementation for potential use in Gateway and other elements of the Artemis mission



Notional NASA Habitation structure near the Moon

Autonomous Avionics Risk Reduction with Lockheed Martin

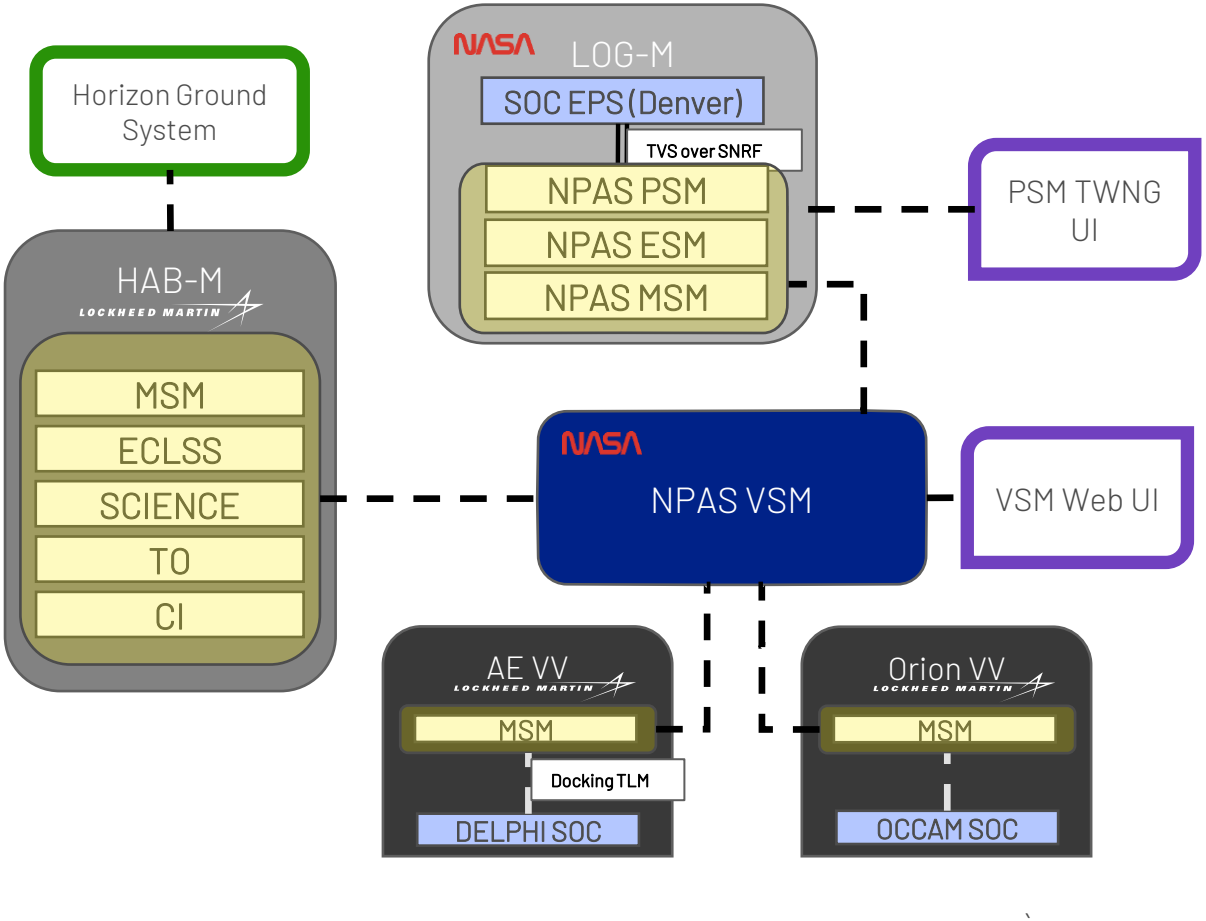


Development and Demonstration Acknowledgement

- Chris. Moore, NASA AES
- Exploration Capabilities Manager and Denise Varga, AES Habitations Support Systems Domain Lead for HQ
- Danny Carrejo, JSC Exploration Integration and Test Lead for facilitating the support at JSC.
- Duane Armstrong, NASA SSC, Test Technology Branch Chief
- D2K Technologies' Quentin Oswald, Joshua Broberg, Brian Rey, Federico Piatti, and Michael Walker
- Lockheed Martin's developers Mike Rosenberg, Cory Kennedy and Jaret Anderson.
- Lockheed Martin's Horizon team including Al Faymore, Christian Morrow, and John Slezosky
- Lockheed Martin's Angie Kibler and Matt Goman.

Risk-Reduction Autonomy Implementation to Enable NASA Artemis Missions

Vehicle System Manager / Module System Manager



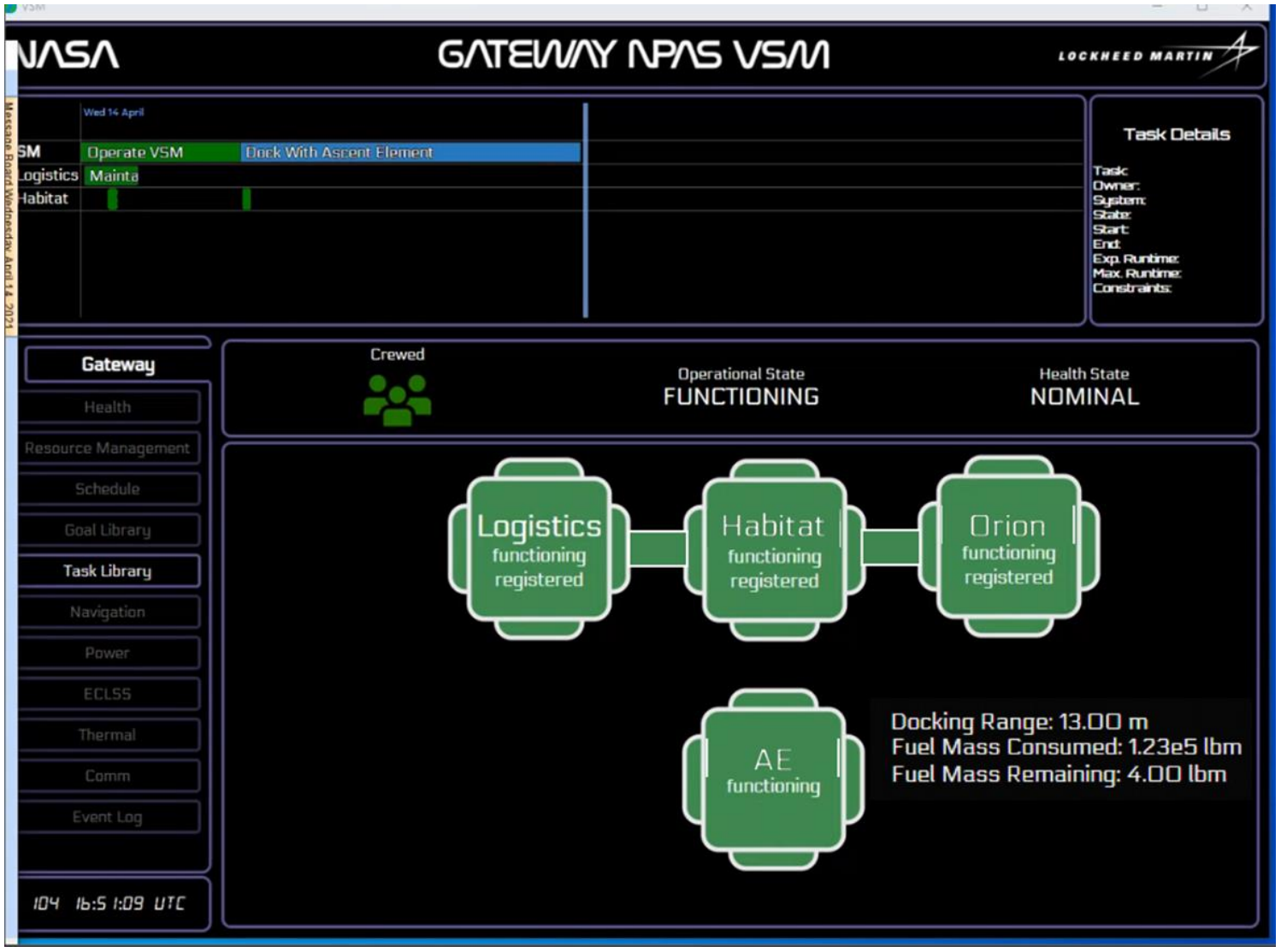
AE Docking and Registration with NPAS VSM

- AE receiving telemetry from DELPHI-29 for docking
- AE docking hold – docking hold, release when ready to register module
- AE docking countdown / docking complete
- AE registration with NPAS VSM
- NPAS VSM displays AE telemetry
- NPAS VSM displays AE's registered tasks

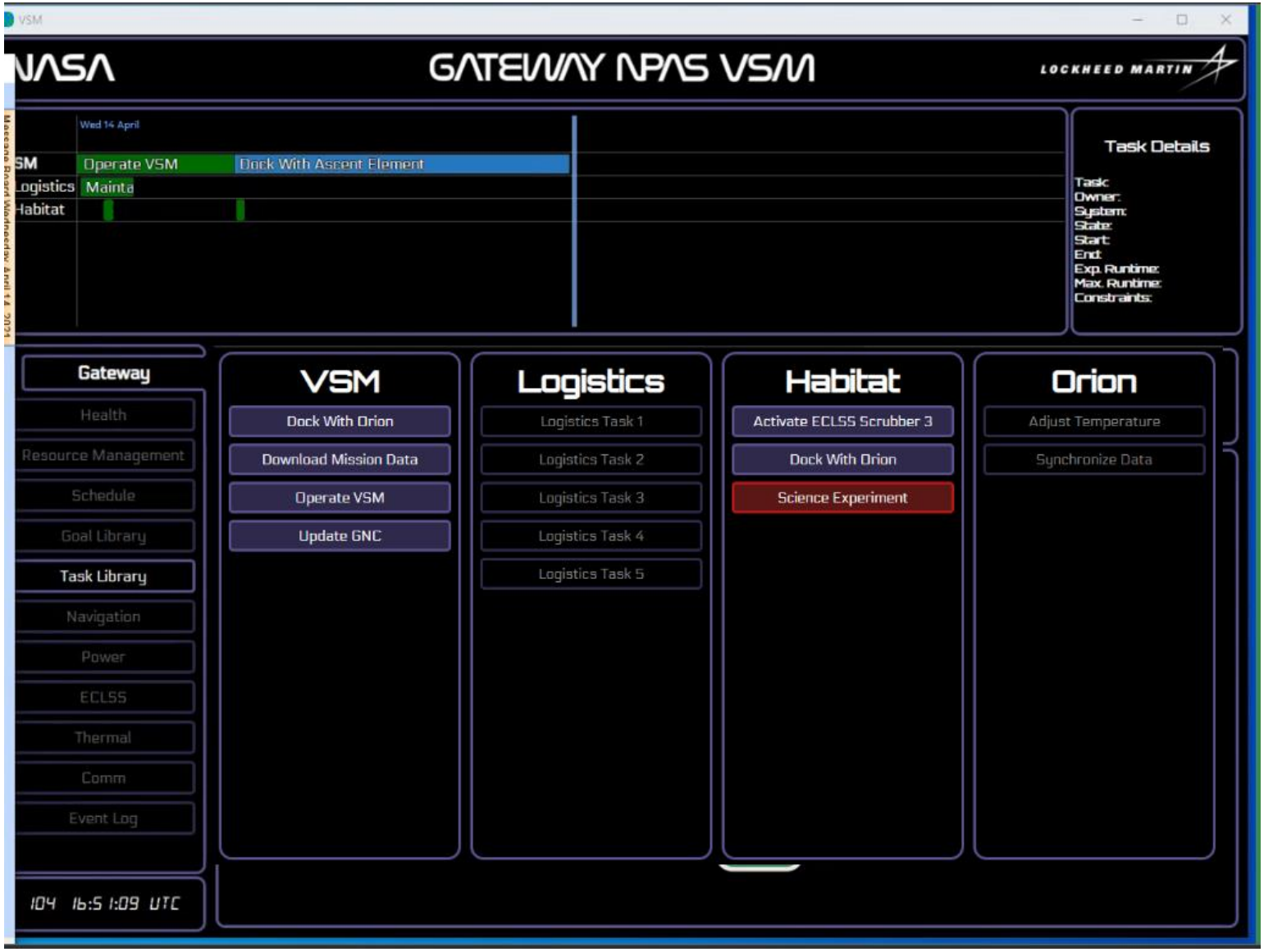
Logistics Module Fault Isolation

- LOG-M Power-SM (PSM) receives telemetry from EARHART (SOC EPS) – Denver
- PSM and ECLS-SM (ESM) detect faults
- PSM and ESM diagnose fault, determines fault can't be isolated, and sends assessment to VSM
- VSM displays fault details
- VSM instructs LOG-M to execute a timeline of recovery tasks on PSM and ESM
- PSM determines the fault is not in the power system and informs VSM

VSM Crew Interface

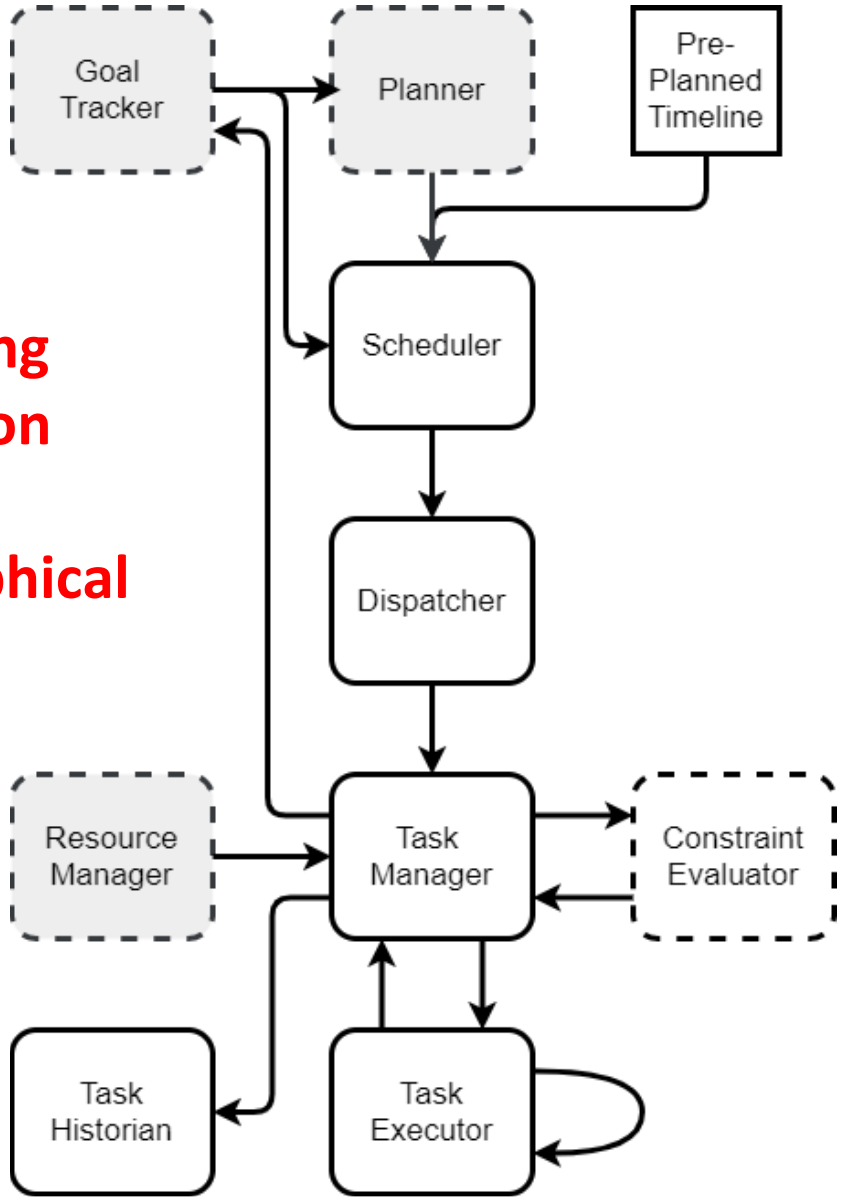


Task Library



Timeline Management and Task Execution

Trust from compartmentalizing functions of mission management and generation of graphical code for visual assurance



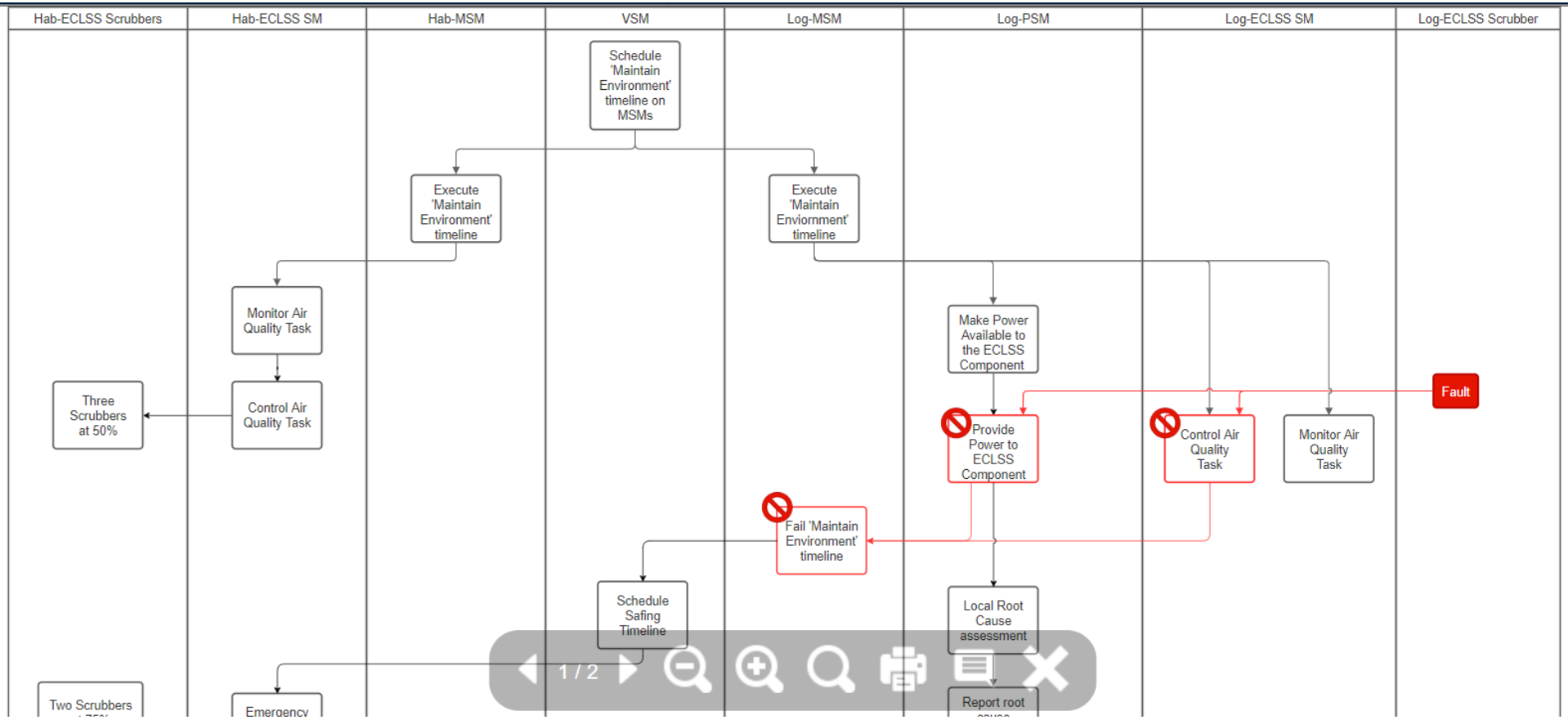
TASK AND TIMELINE DEVELOPMENT AND EXECUTION

The VSM's tasks and timeline execution is modeled after the NASA Gateway autonomy operations timeline and mission management framework. The layout of the framework consists of the following components:

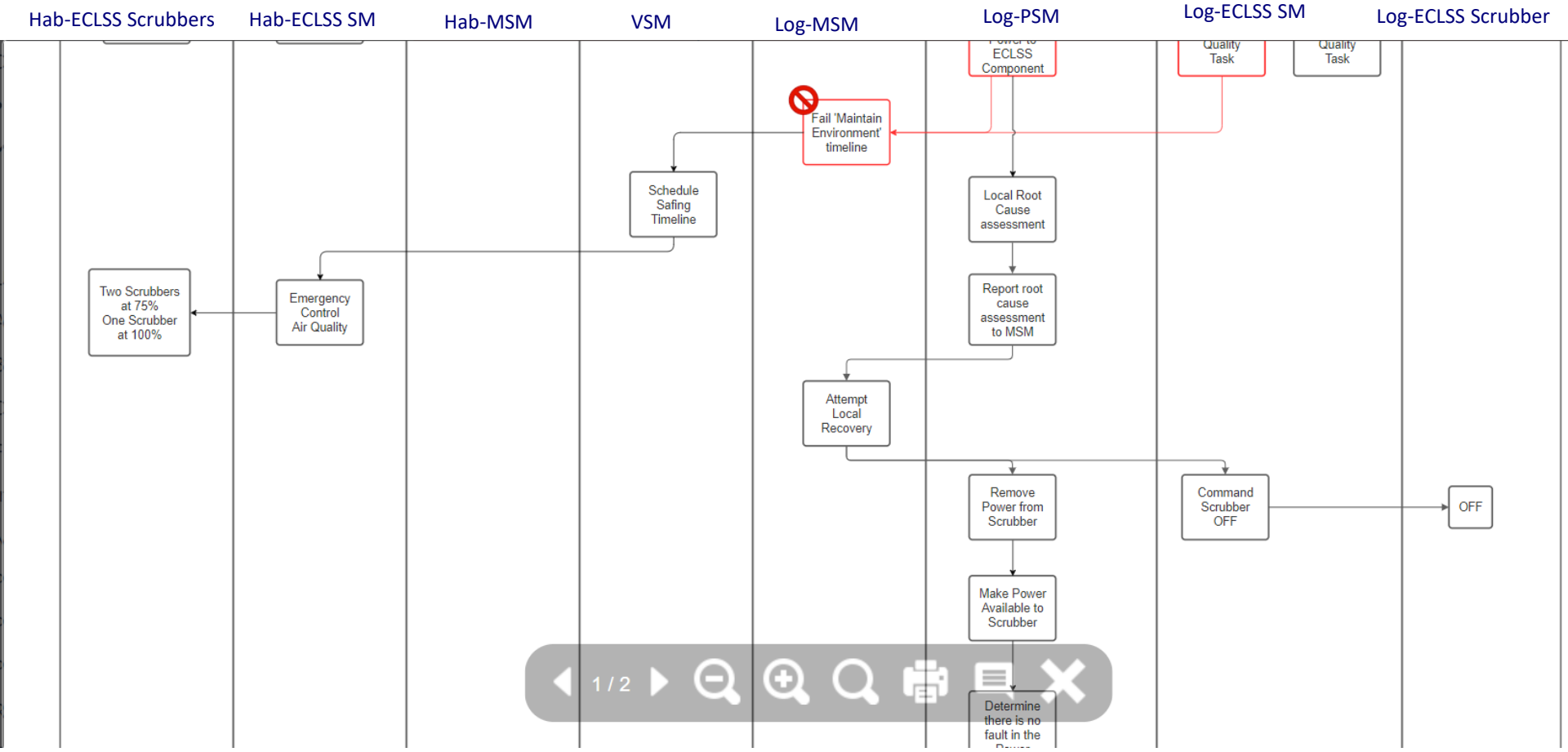
- **Planner**: Determines the set of tasks and their order of execution that is required to transition the system from the current state to the desired goal state. The Planner sends the ordered blocks of task to the Scheduler for scheduling. Failed tasks cause the Planner to re-plan based on the current conditions.
- **Goal Tracker**: Monitors and records the status of system's progress toward goals.
- **Scheduler**: Receives blocks of planned tasks from the Planner and schedules them into a timeline based on timing constraints, operational constraints, system health conditions, and resource availability. The resulting timeline is segmented and sent to the Dispatcher.
- **Dispatcher**: Manages the scheduled operational timeline of tasks. The Dispatcher is responsible for starting tasks at their scheduled time and merging new timeline segments received from the Scheduler into the operational timeline.
- **Task Manager**: Manages the execution of tasks that the Dispatcher has started. The Task Manager uses the Constraint Evaluator to verify preconditions have been met, periodically checks for invariant violations, and monitors lapsed execution time for constraint violations. The Task Manager receives periodic task status updates from the Task Executor while a task is running and reacts to its status — either aborting the task if a constraint is violated or notifying the Goal Tracker and Planner of success. The Task Manager always notifies the Task Historian regardless of the task's outcome.
- **Task Executor**: Executes the task provided by the Task Manager and periodically reports the task's status to the Task Manager. A Task Executor can execute the task within its local system or forward it to lower level Autonomous System Manager. For example, a MSM would forward a power system task to its PSM.
- **Task Historian**: Receives and records tasks as they are completed — successfully or unsuccessfully — by the Task Manager. The records serve as an audit trail for the timeline execution.

At the current stage in development, the timeline segments provided to the Scheduler are pre-defined, emulating the uploading of schedules by the ground crew to the VSM. Because of this, the Planner and Goal Tracker components have not been incorporated into this implementation. The Resource Manager and Constraint Evaluator are also rudimentary and will be improved in future releases.

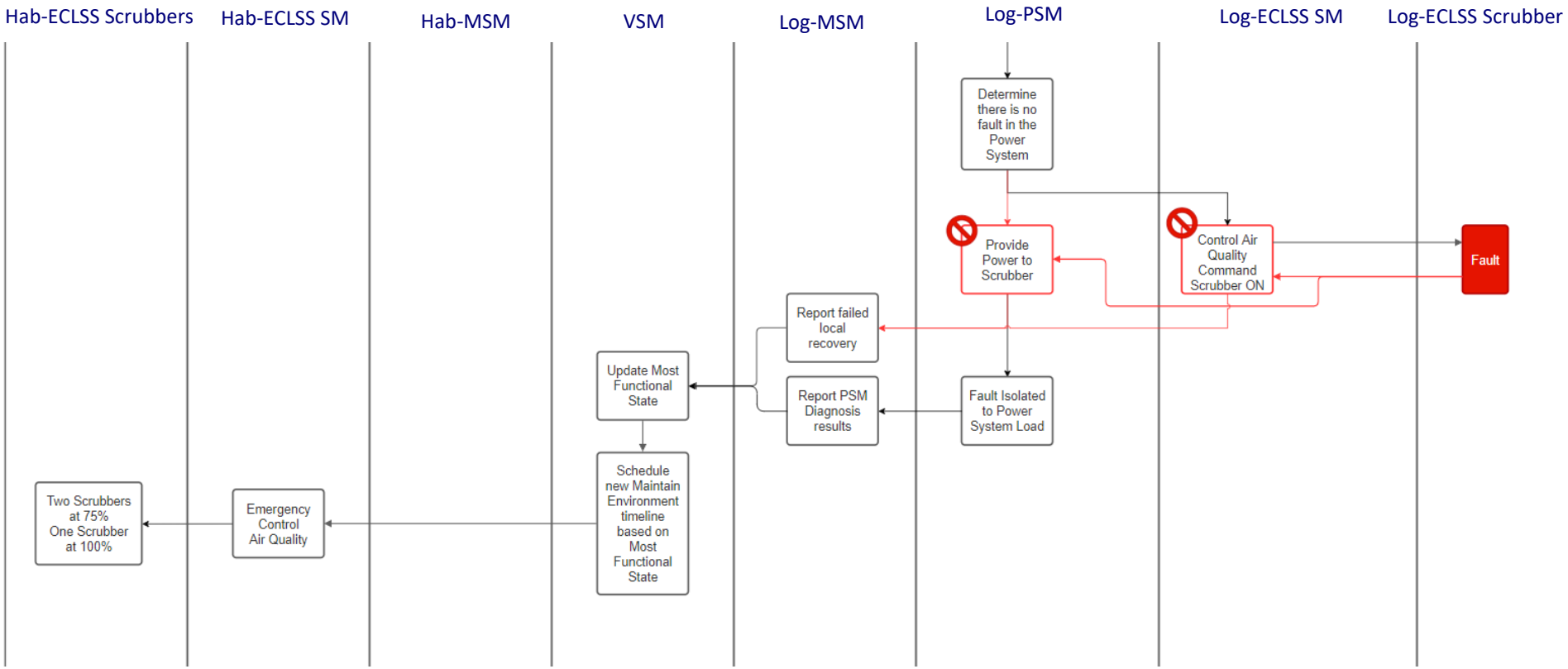
Scheduling and Executing Tasks Encompassing a Failure in Logistics Module ECLSS



Scheduling and Executing Tasks Encompassing a Failure in Logistics Module ECLSS (Cont.)

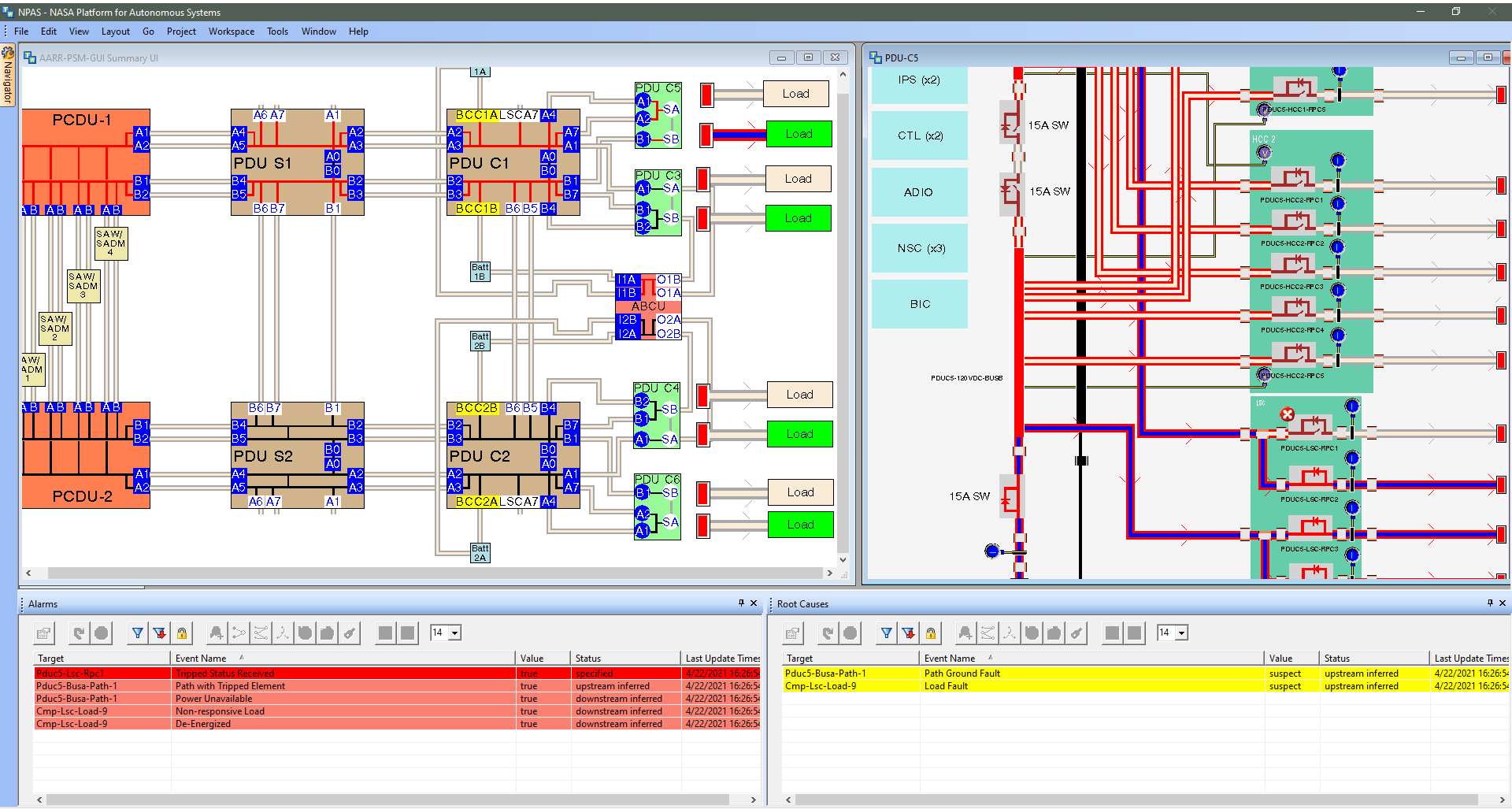


Scheduling and Executing Tasks Encompassing a Failure in Logistics Module ECLSS (Cont.)



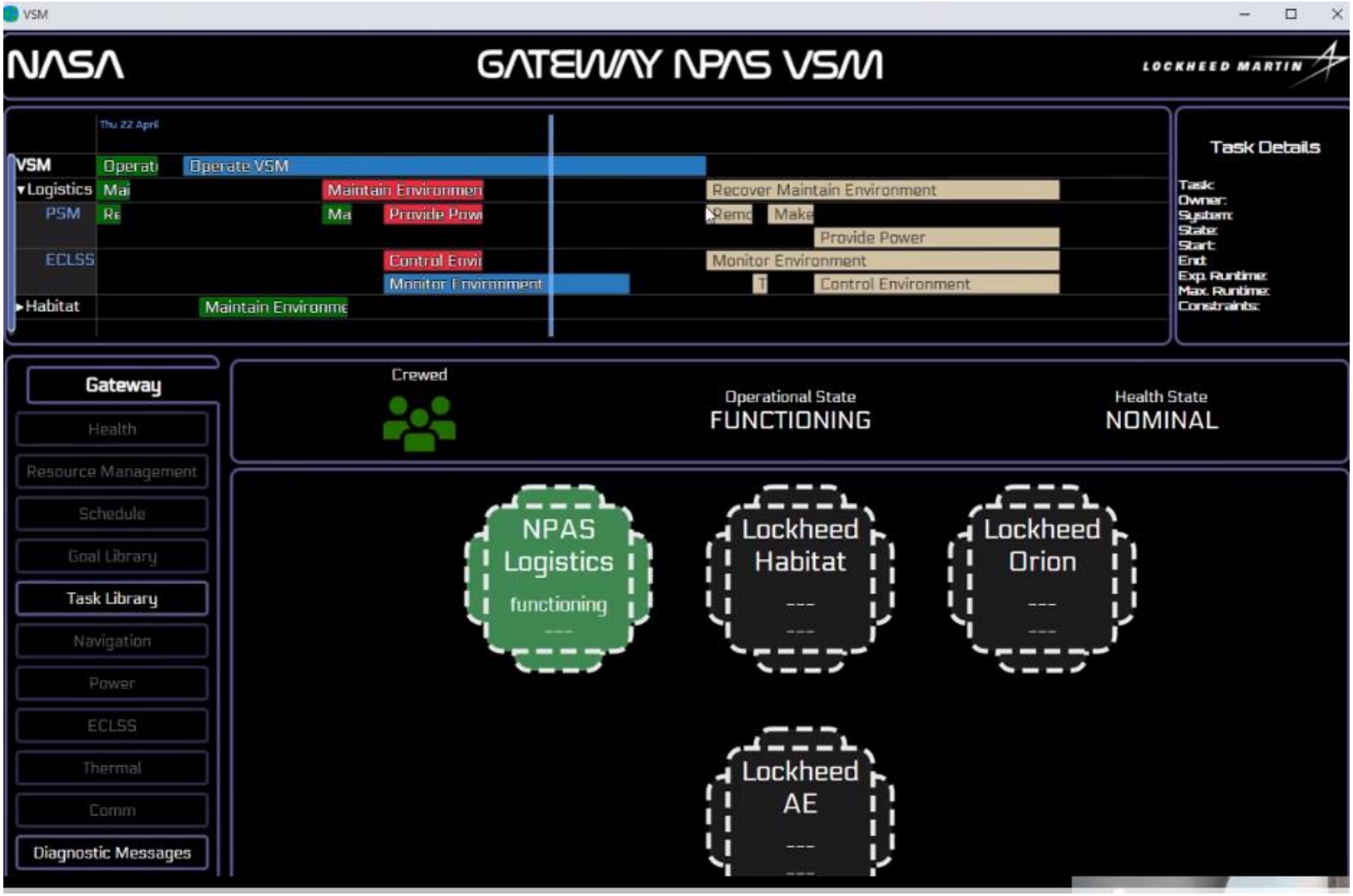
Fault Suspects

Stennis Space Center



Trust from digital-twin awareness of as-is system while in operation

NPAS VSM Timeline Task GUI Showing Failed Tasks

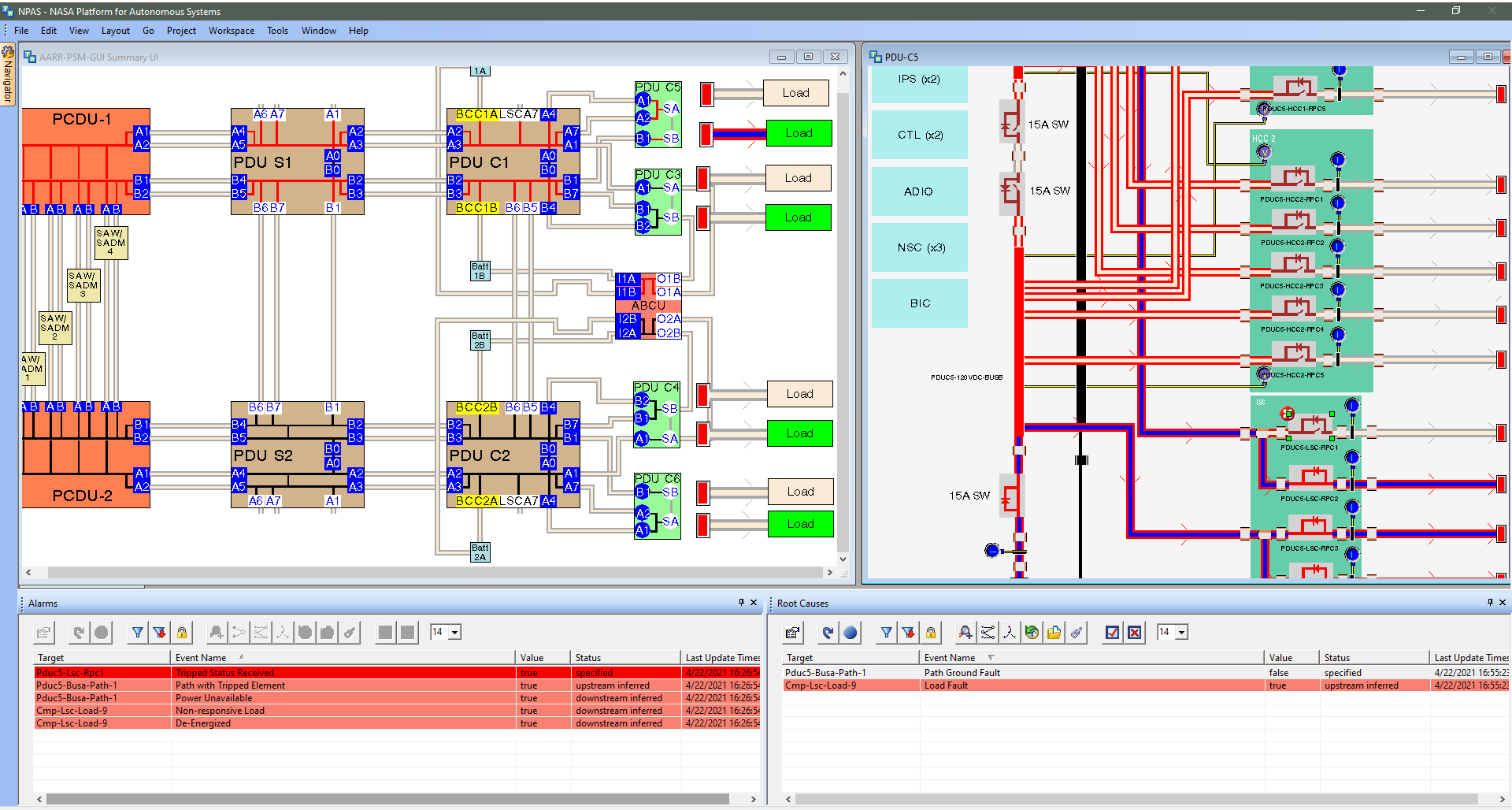


Trust from visual timeline execution awareness

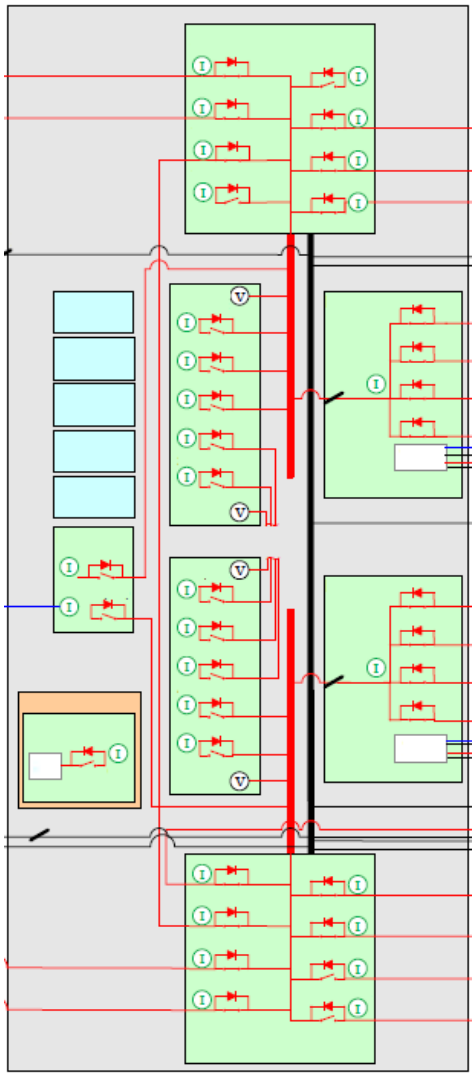
Safing and Recovery

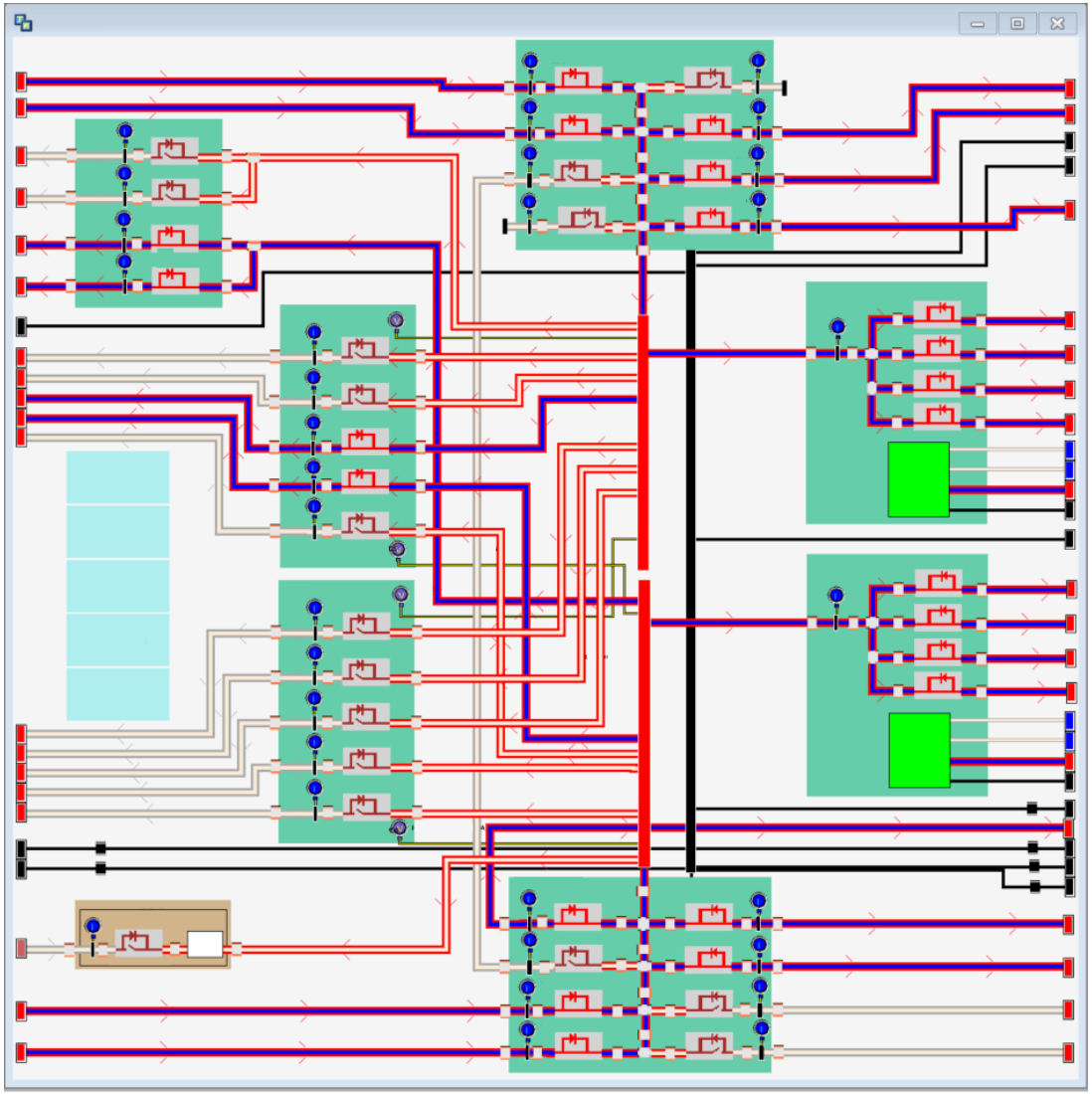
- VSM has Safed the vehicle by scheduling the Emergency Control Air Quality task on the Habitat MSM - this task increases the scrubber output to compensate for the failure and
- In parallel VSM has scheduled recovery tasks for Logistics that step through an automated troubleshooting process to isolate the source of the fault to one of the Systems.
  - First Logistics MSM ensures the scrubber is unpowered and commanded off
  - Then PSM energizes the scrubber's distribution path and tests for a ground fault; no fault is found and this new evidence is added to PSM's fault diagnosis, shown as a green entry the message log
  - Next the Logistics MSM commands ECLSS to turn the scrubber ON and PSM to provide power to it.
  - The overcurrent condition happens immediately when the Scrubber is turned ON; indicating that the fault is in the ECLSS component and not the Power System.
  - PSM's fault diagnosis is updated with this new evidence; leaving only one possible root cause for the failure, shown as the red entry in the message log.

Fault Isolation



Real-Time Life Schematic

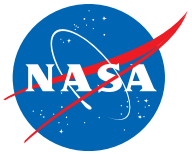




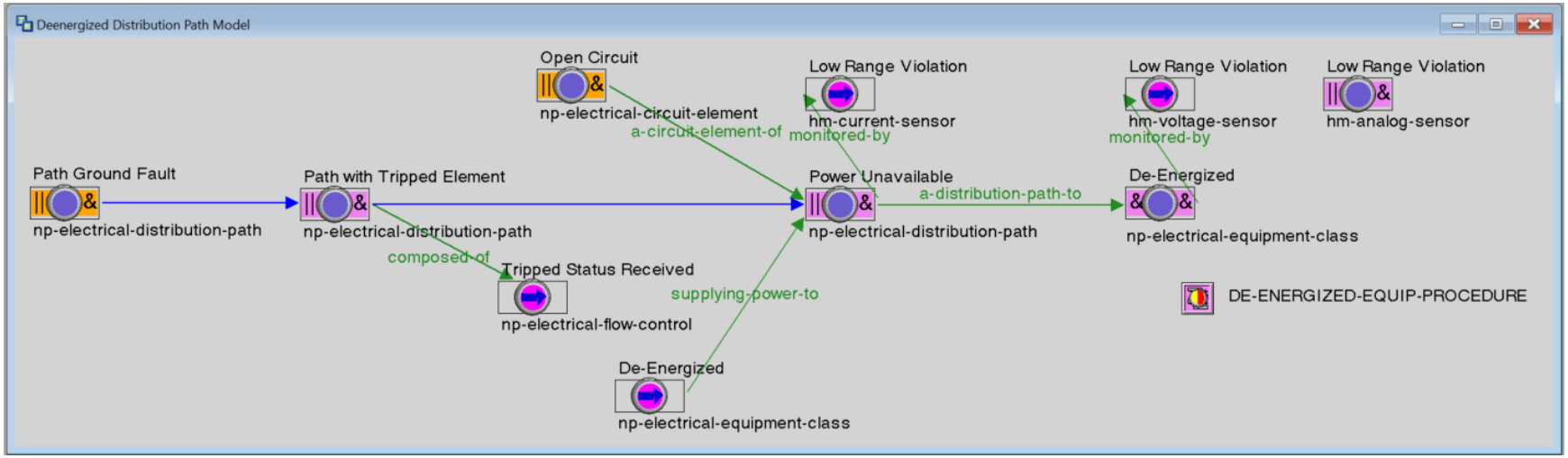
Most Functional State

- After the attempted recovery of Logistics capability VSM can no longer schedule the failed Logistics Module's maintain environment tasks so Gateway has a new Most Functional State and
- VSM has scheduled a new timeline to maintain the environment using Habitat Modules' Emergency Control Air Quality tasks with the increased ECLSS output and only uses Logitics ECLSS to Monitor the environment,.
- Going over to the Horizon display we can see Habitat MSM the increased scrubber output from the nominal 3 scrubbers at 50% to 1 scrubber at 100% and 2 at 75%

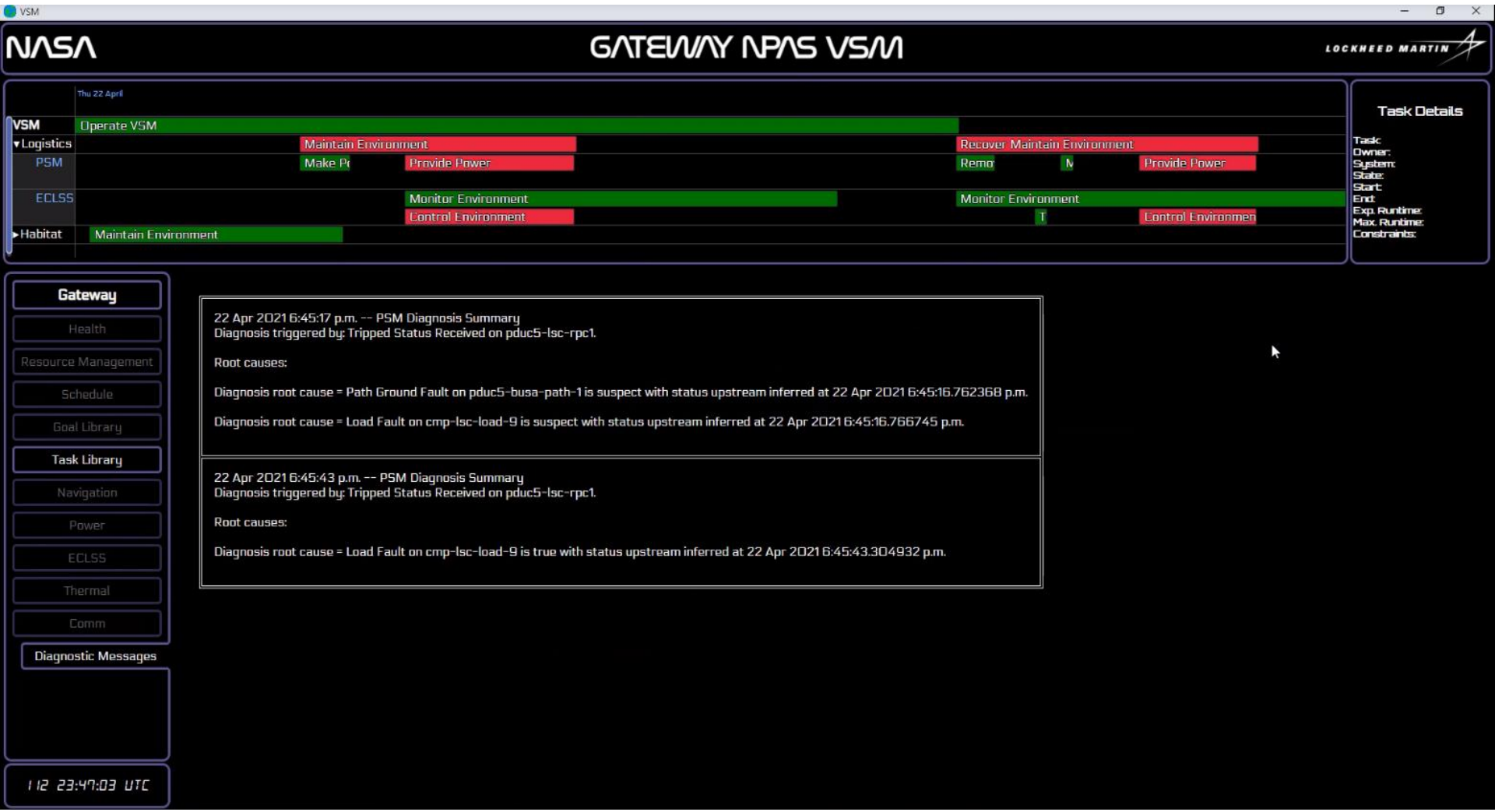
Additional Slides



PSM Generic FMEA with RPC Trip Event



VSM FMEA Report



NPAS - NASA Platform for Autonomous Systems
Roadmap


UPSS @ KSC


Orion EFT-1


RAD750 (SP-0)




SVMF @ JSC – HITL with NextSTEP-2 partner

2005 Intelligent ISHM Initial funding for integrated systems health management, testbeds and prototypes

2006

2007 Public-Private Partnership with General Atomics for an ISHM integration & modeling software platform

2008

2009 Chemical Steam Generator pilot implementation

2010

2011 CSG Skid Pilot Project

2012 KSC Cryogenic Testbed


CTL @ KSC

2013

2014 Autonomous Propellant Loading Demo at KSC

2015 Universal Propellant Servicing System at KSC

2016 Orion EFT-1 Demo

2017 HPGF ISHM Monitoring System

Autonomous Systems Domain Specific Language

Port G2 to RAD750

Demo at JSC of Distributed Autonomous Operations – Vehicle Manager, Power & Avionics Systems

2018 INSIGHT Nitrogen Verification, Validation & Deployment

Autonomous Systems Design Guidelines for Flight Hardware

2019 Autonomous Systems Intelligent Strategies

SSC-MSFC partnership NPAS certification for Class A systems

NPAS optimization for flight systems via public-private partnership. SBN Bridge for seamless NPAS-cFS interface

Gateway demo with NextSTEP-2 partner

2020 Autonomous Avionics Risk Reduction Activity with Commercial Partner


X-Lab @ JSC - autonomy architecture down to the system manager level; create schedule and execute task and timelines across multiple modules developed by separate teams

2021

2022 NPAS demo for Gateway

2023 LSII: Build, and Excavation Autonomous System Transport (BEAST). Lead by GRC



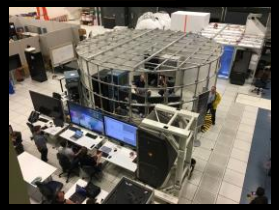
SSC ASL

Establish new capabilities to design, build and deploy intelligent autonomous systems
Enable rapid, economical development of robust, safety-critical autonomous systems
Support ground operations and Exploration missions
Can also benefit SMD and ARMD and HEOMD Tier 1 Capability Gaps
Utilize experienced space systems engineers and s/w developers

Agency Technical Teams

NASA Gateway VSM Working Group
NASA Gateway Autonomous Systems Manager Group
NASA Autonomous Systems Capability Leadership Team
NASA OCT "Study on Operating Autonomous Missions" lead by JPL's CCT
NASA Space Trusted Autonomy Science and Technology Partnership with US Space Force and NRO
Lead of the Avionics Instrumentation Community of Practice, supporting the NESC Avionics Discipline Expert
NASA Gateway Test and Verification Working Group
Gateway Modeling and Simulation Working Group
Subtopic Manager SBIR Topic "Autonomy Technologies for the Deep Space Gateway"
Subtopic Manager STTR Topic "Autonomous Systems and Operations for the Lunar Orbital Platform-Gateway"
Autonomous Systems and Robotics Strategic Capability Development Plan


HPGF @ SSC


iPAS lab @ JSC


iPAS lab @ NPAS demonstrated as an integrated hierarchical distributed capability

NPAS Implementations FY22

- AES Polaris Project: Autonomous Satellite Technology for Resilient Applications (ASTRA). Partnership with Sidus Space
- AES CHP EVA: In support of Crew Health and Performance (CHP) EVA (Extravehicular Activity) , NPAS will coordinate with the Human Physiology, Performance, Protection & Operations (H-3PO) Lab – Space Suits and Exploration Operations Team (SSEO)
- CIF (STMD) Enhanced Autonomous Refueling Capability for Gateway and Surface Systems. Collaborating with Gateway Chem Prop Breadboard Team (JSC), following GATEWAY SUBSYSTEM SPECIFICATION FOR REFUELING (GP 10126) and GATEWAY REFUELING CONCEPT OF OPERATIONS (GP 10140)
- STMD LSII Project BEAST (Build & Excavation Autonomous System with Transportation) lead by GRC (cancelled due to budget reductions)

PROJECT POLARIS

ASTRA

Stennis Space Center

Project Description & Objectives

Title: Autonomous Satellite Technology for Resilient Applications (ASTRA)

Description: ASTRA will demonstrate an AES-derived autonomous operations technology in a spaceflight environment, provide flight heritage and testing for an autonomous system manager platform.

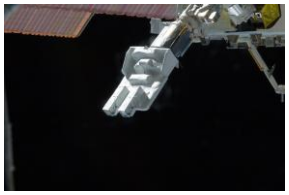


Objectives: Evaluate, validate and stress test on-orbit, autonomous operations of satellite management and system functions.

Detailed Project Overview

- Include an AES autonomous operation technology investment on LizzieSat™ (a Sidus proprietary multipurpose satellite bus that integrates custom payloads).
- Leverages contract Sidus has with NASA to provide services to manage, integrate, deploy and perform on-orbit operations of commercial satellites (under 110kg), into low Earth orbit, using the Space Station Integrated Kinetic Launcher for Orbital Payload Systems (SSIKLOPS) from the International Space Station to deploy LizzieSat™.







Technology Infusion & Innovation Approach

Infusion: Develop an autonomous system manager for satellite operations including Integrated System Health Management (ISHM), resource management, vehicle management, subsystem management and, ending with a flight demonstration of autonomous satellite operations using the NPAS platform

- ✓ Demonstrate ability for autonomous operation of satellite systems
- ✓ After establishing confidence, fly vehicle/satellite autonomously near end of life of satellite

Benefits of Approach:

- ✓ Provide affordable on-orbit testing demonstrating autonomous systems ISHM/FDIR capabilities during satellite operations.
- ✓ Conduct a systematic on-orbit stress test of autonomy, using satellite system data and simulated anomalies, to establish boundaries for trusted autonomous operation.
- ✓ Advance TRL, and establish flight heritage for autonomous operations, to mitigate barriers to adoption by future Artemis missions and commercial partners

Project addresses **Tier 1 gaps, in TX: Tech Area 03, Power and Energy Storage, 10, Autonomous Systems & 13, Uncrewed Surface Systems, required for sustainable foundational systems**

- ✓ NASA has IDIQ contract with Sidus to provide turnkey services to manage, integrate, deploy and perform on-orbit operations of commercial satellites into low Earth orbit (LEO) using the Space Station Integrated Kinetic Launcher for Orbital Payload Systems (SSIKLOPS).
- ✓ Sidus will leverage this contract to deploy LizzieSat™/small satellite, proprietary multipurpose satellite bus that integrates custom payloads, from ISS utilizing SSIKLOPS, and conduct all mission operations
- ✓ The ASTRA team will work with Sidus Space to integrate the payload on a small satellite.
- ✓ The ASTRA team will design, build, and test the ASTRA autonomous systems hardware and software.

References to NPAS and Recent Projects

TechPort Fact Sheet: https://techport.nasa.gov/view/94884

Sidus Space Press Release: https://sidusspace.com/09-2021-sidus-space-awarded-nasa-heomd-aes-project-polaris-awards-for-autonomous-satellite-technology-for-real-time-applications-astra/

NASA Internal – Project Polaris Press Release: https://www.nasa.gov/feature/nasa-empowers-workforce-to-advance-deep-space-technologies

D2K Press Release: https://www.d2ktech.com/post/nasa-aes-awards-contract-for-autonomous-systems-in-space

NASA SSC Press Release: https://www.nasa.gov/centers/stennis/news/releases/2021/SSC-Gaining-Recognition-for-Cutting-Edge-Autonomous-Systems-Work

NPAS Concluding Statements

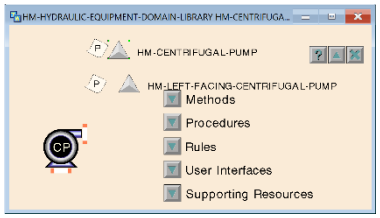
- ✓ NPAS is an innovative software platform that provides the framework to quickly develop, implement and deploy intelligent, autonomous operations software
  - integrated architecture for creation and analysis of a comprehensive “live” application model, health assessment, diagnostics, anomaly detection (FMEA), autonomous operational strategies, mission planning, scheduling, and execution, as well as network communication for system integration
- ✓ NPAS has reusable core capabilities
  - Demonstrated NPAS s/w and models are easily reused for other missions
  - reduced development time and cost
- ✓ Evolvable/scalable
- ✓ NPAS is certified for Class C, Safety-Critical operations
- ✓ Targeting a FY22 CMMI Appraisal for Class A human rated software
- ✓ NPAS runs on current spaceflight processors (SP0-S) in VxWorks, and on COTS versions of ARM processor that are flight qualified
- ✓ NPAS can fully integrate with cFS software via the G2-SBN Bridge



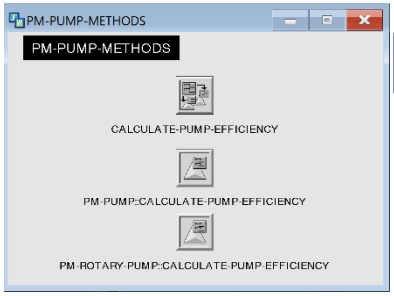
Back up

NPAS includes physics based models

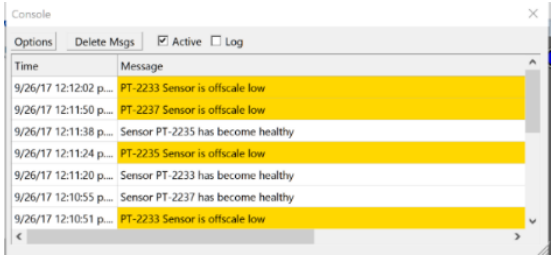
Stennis Space Center



Definition of a class for a rotary pump

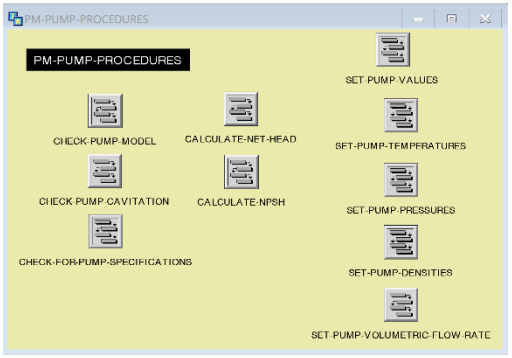


Calculating pump efficiency



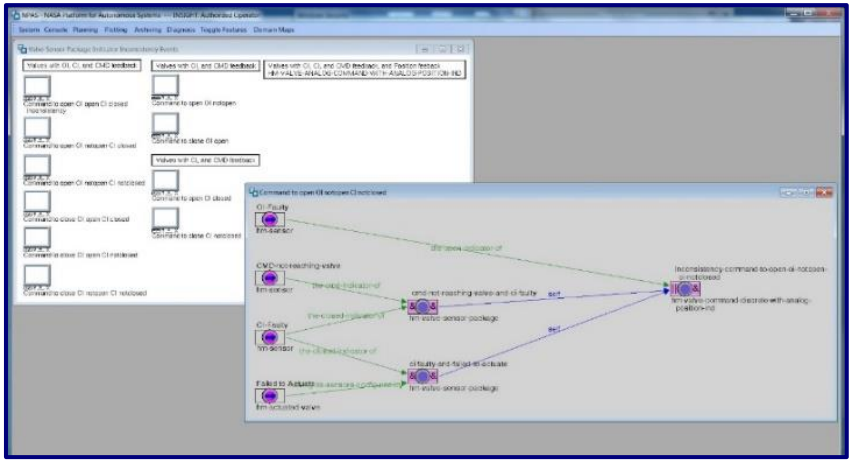
For every rotary pump instance

Example, of an inconsistency of a sensor, and this inconsistency is triggered by NPAS models being applied; physics, or other models in NPAS are detecting events; these events are used in root cause trees for diagnosis (as shown in example below)



Checking consistency of models against real-time data

NPAS is applying models comprehensively throughout the system periodically or based on triggers



Root cause tree associated with a valve assessment model

Execution of models (blocks above) generate truth values on events that form part of cause-effect trees (on the right), corresponding to FMEA and/or used for other reasoning as needed by autonomous operations activities

NPAS autonomy strategy: Use replacement for failed sensor

- Example concepts that NPAS understands
  - Piping section
  - Membership in a piping section
  - Sensor classes/function i.e. thermocouple, pressure sensor
  - Replacement sensor (a kind of redundant sensor-that is in the same piping section as a sensor that has failed)
  - Sensor health (availability)
- Autonomy strategy
  - If a redline sensor becomes unhealthy, look for a replacement sensor; if one exists, continue operations using the replacement sensor
  - If none exist, stop the sequence (if one is being run) and inform the operator (or a sequence for safing the system could be activated)

AL Monitors - SELECT STATUS TO MANUALLY ACTIVATE/DEACTIVATE --- SELECT PLAN TO SELECT LOAD PLAN OR SET TO NO PLAN

	Status	State	Alternate	Plan Assigned - Red/Yellow if Red Line, White if Normal	Triggered
PT501	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	7/24/14 4:15:05 p.m.
FT103	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	Not Triggered
DP212	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Adv_t...	9/16/14 4:16:43 p.m.
PT199	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Adv_t...	Not Triggered
VFD3-AMPS	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	Not Triggered
PT173-PT161	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	Not Triggered
SAT_PT161	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	9/19/14 12:29:20 p.m.
PT161	Deactivate	HEALTHY	UNKNOWN	C:\ishm-nasa-toolkit\sequence-plans\Automatic Demo Plans\Modifi...	9/16/14 6:39:25 p.m.

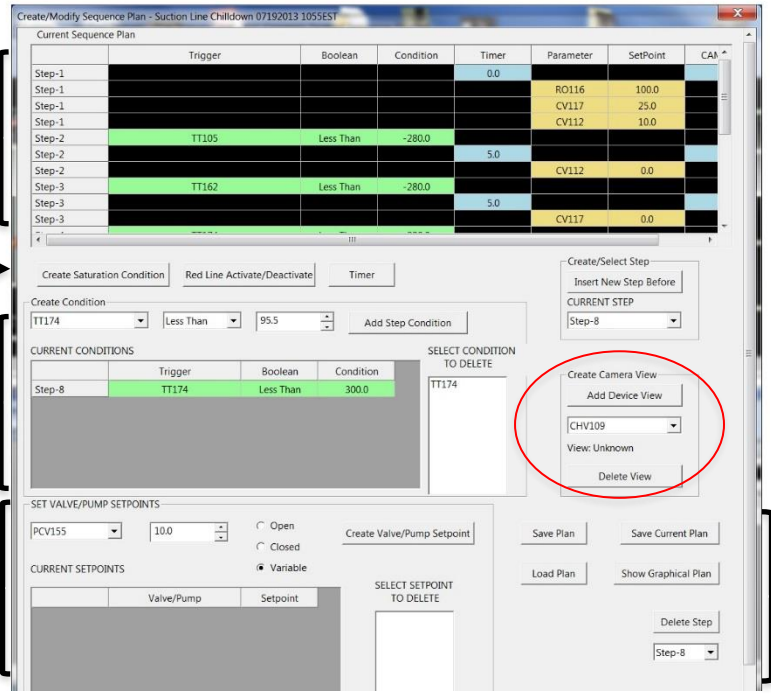
Critical sensors monitor showing “alternate” sensor availability should they fail.

Note: This autonomy strategy is “generic”/reusable and applies to any system – no specific coding is needed to apply to another system

NPAS autonomous sequence operations

NPAS enables quick and easy creation of sequences using menus

- Sequences
  - represented in tabular and graphical formats
  - verified by simulating conditions that enable actions to be executed
  - missions defined by multiple interacting sequences
- Sequence conditions include:
  - sensor triggers
  - redline triggers
  - fluid saturation state
  - redline sensor failure
  - may include other conditions from health or other algorithm outcomes
- Sequence actions include:
  - valve and pump operations
  - camera pointing
  - execution of special sequences i.e. as shut-down or reverting to a prior step
  - can include execution of any activities that responds to system conditions and planning



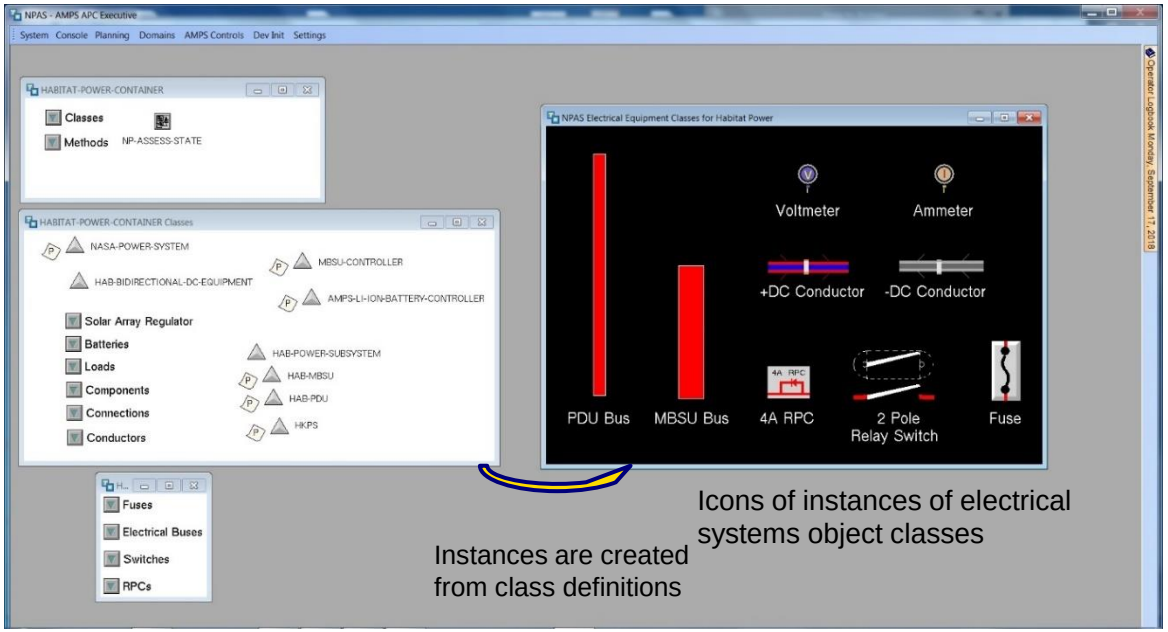
Sequence creation, loading, and execution

…. sequences are fundamental building blocks of plans and activities

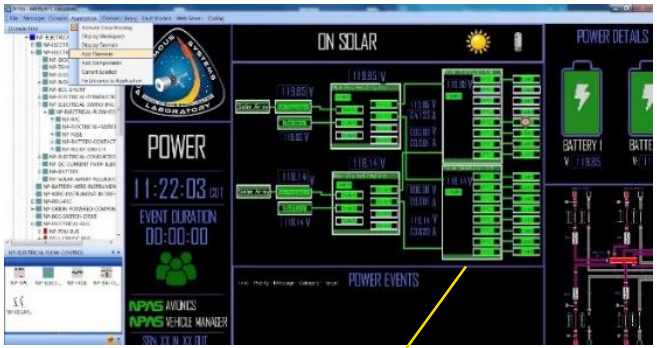
NPAS palette use: power example

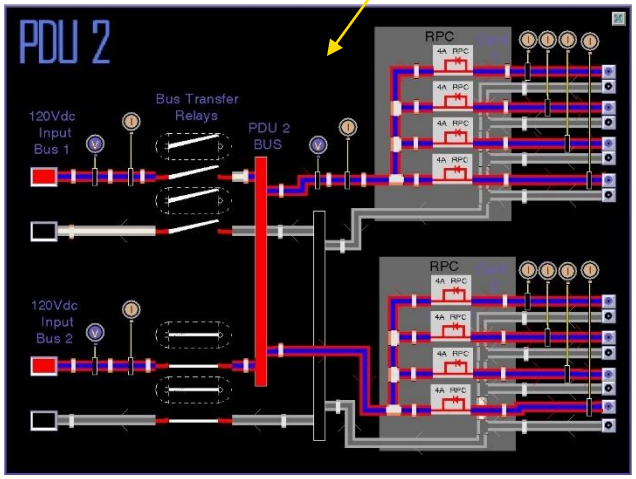
Stennis Space Center

Power equipment classes with sample instance



Power UI with palette





Power system domain model based on schematics

Class definitions, used to create an instance

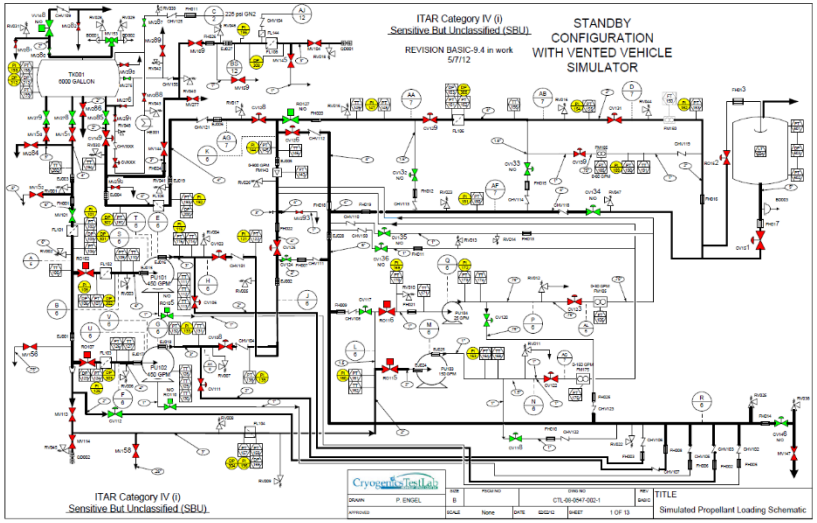
Graphic-oriented environment

- create the model of the application graphically (schematic)
- objects are represented by icons
- objects-representation of some part of the application
  - i.e., tank, valve
  - represented by an icon
  - has a table of attributes (contains the knowledge about the object)
  - Hierarchy of parent and child classes of objects
- objects are placed in workspaces- i.e. like a diagram in SysML
- objects are connected graphically
- pop-up menus for objects (i.e. attribute table)
- actual application objects = instances

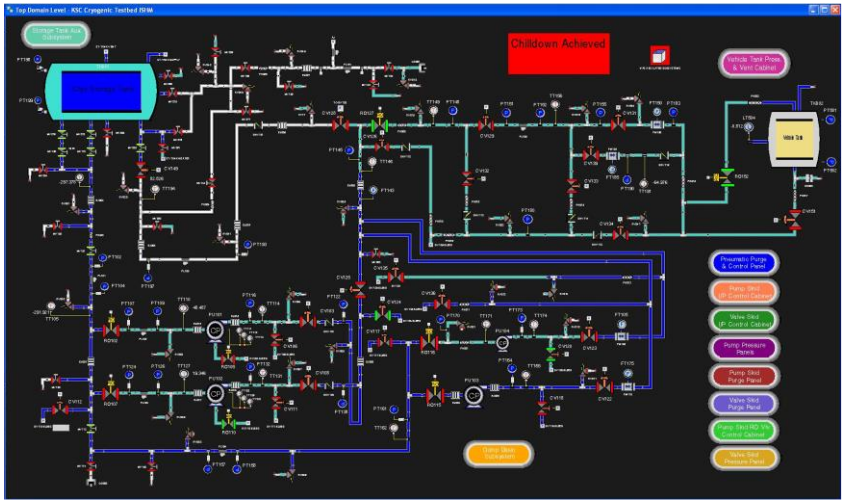
Instances are connected according to schematics to create knowledge domain models

Getting started in NPAS

- Start with system schematics and ConOps
- Create NPAS instance of every object in schematic and link them together (use library of palettes)
- Build NPAS “schematic.” This creates the core application knowledge domain – how the knowledge base is formed
- Use this information for reasoning & discovery (NPAS ISHM and Autonomy functions are embedded in NPAS and will apply to the domain from the schematics)



Schematic, KSC Cryogenics Test Laboratory (CTL) – old version



Application domain model, KSC CTL

INSIGHT

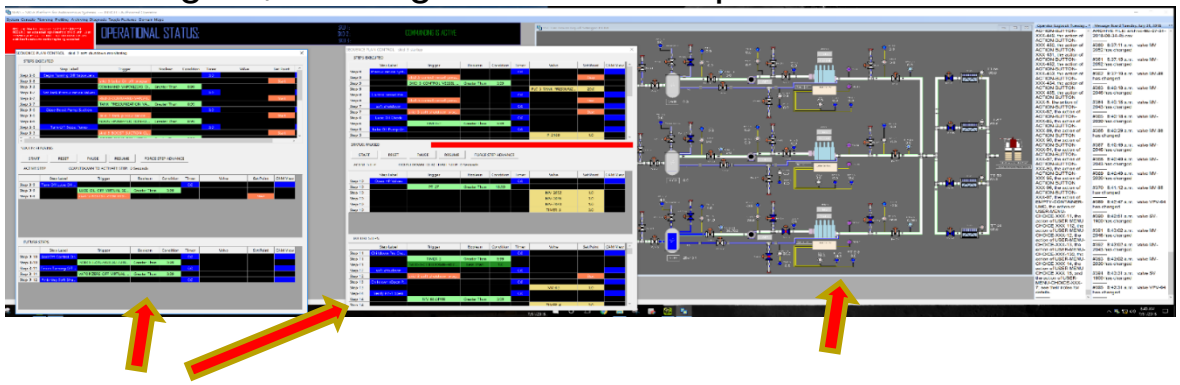
Stennis Space Center

Intelligent Stennis Gas House Technology at SSC's High Pressure Gas Facility (HPGF)

NPAS created a digital twin to enable intelligent, thinking autonomous operation

Checkouts: one of the first steps performed by the system is to verify that the safety shutdown system is operating properly.

The graphic illustrates the redline checkout performed for test TPS-HPGF-01350 on July 31st, 2018, prior to Pump Sequence Operations on Skid 3



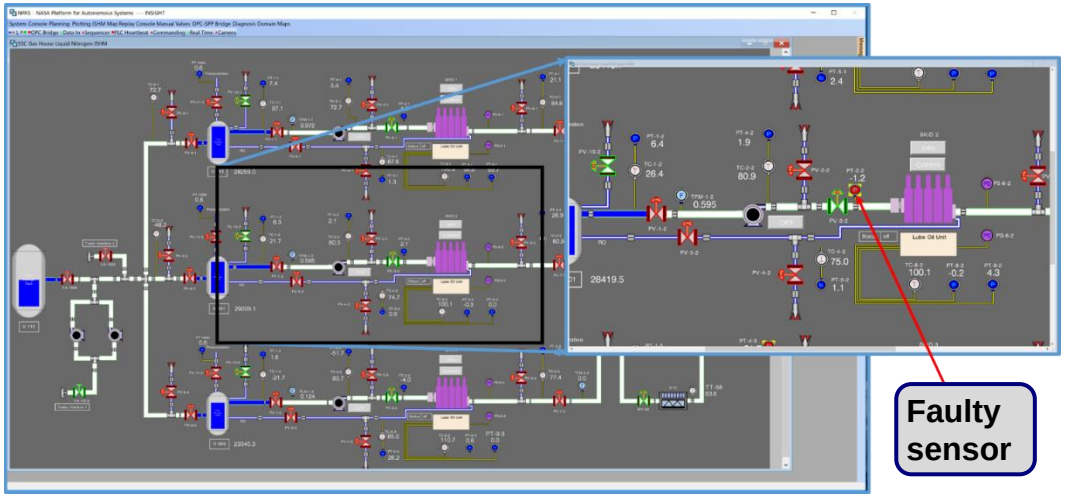
- Graphics windows developed with NPAS for Project INSIGHT to illustrate the sequence that is operating in real-time
- Displays both previous sequence steps as well as upcoming steps and expected stimuli

Graphics window developed to mimic existing PLC screens -- minimizes operator training time, maximizes familiarization by existing operators and provides a versatile tool to visualize the system, **in real-time**

Class C Safety Critical

Capable of monitoring and controlling the HPGF N2 skids and existing vaporizers

User friendly interface



Faulty sensor