Autonomous Control for Rocket Launch Systems

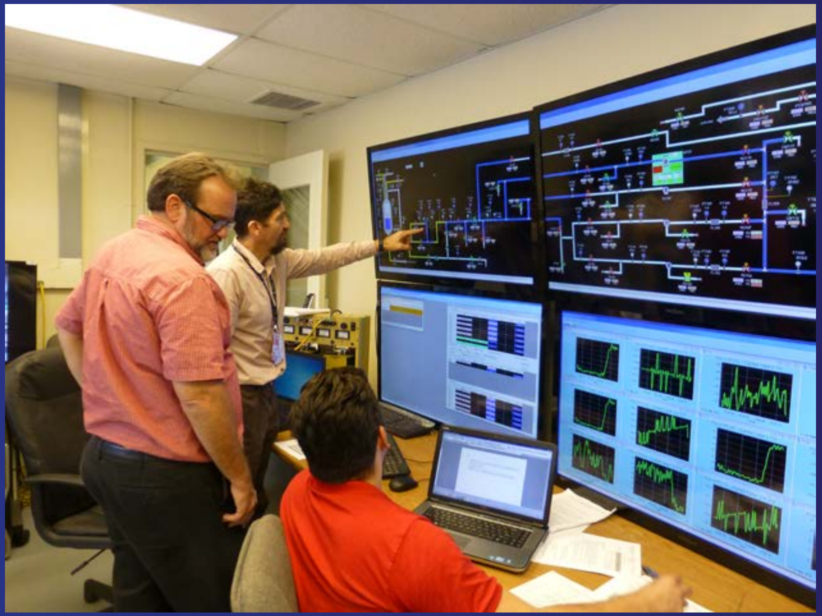
Commercial and Government Response Access to Space Technology Exchange

June 22-25, 2015
Westfields Marriott Washington Dulles
chantilly, Virginia

Fernando Figueroa, NASA Stennis Space Center
Jaime Toro-Medina, NASA Kennedy Space Center
Mark Walker, D2K Technologies
Kim Wilkins, General Atomics
Robert Johnson, NASA Kennedy Space Center
Jared Sass, NASA Kennedy Space Center
Justin Youney, NASA Kennedy Space Center
Gerald Stahl, NASA Kennedy Space Center

Autonomous Control for Rocket Launch Systems

Acknowledgements: Funding for this work was provided the NASA's Advanced Exploration Systems (AES) Division of the Human Exploration and Operations Mission Directorate.



Introduction

- **Autonomous Control (AC) refers to control actions of a system that take place without intervention from humans.**
- **AC denotes control actions that respond to events that are unexpected, and enable the system to continue on a path to achieve an original objective or alternate objectives.**
- **Autonomous Control incorporates concepts such as adaptation, mitigation, and re-planning in space and time.**

Paradigm for Autonomy

- Autonomy is a capability that is not absolute. There are degrees of autonomy, ranging from low levels to high levels, but there is no maximum level (how many autonomy strategies are implemented?).
- It is an evolutionary capability that can handle increasing degrees of complexity for reasoning and decision making.
- It must know the condition of the system elements and their ability to carry out the task. Integrated System Health Management (ISHM) then becomes an enabler for autonomy.

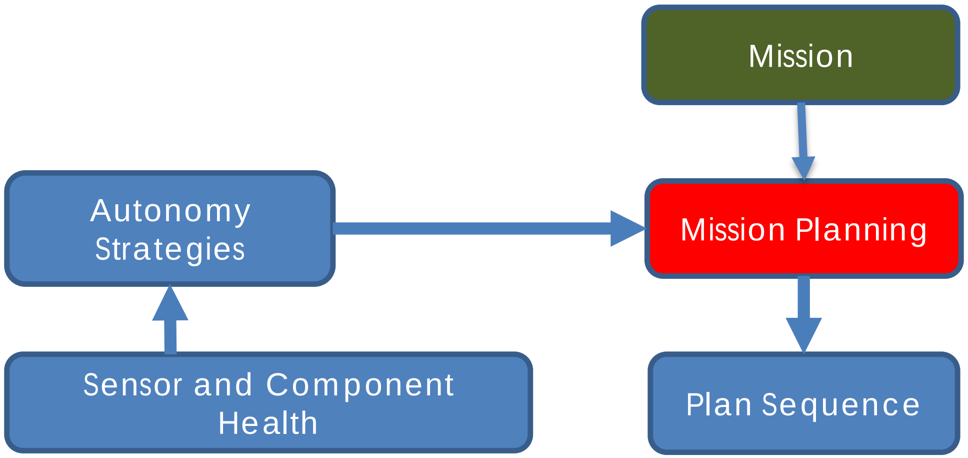
Autonomy Strategies

- Strategies for autonomy guide the decision making process. What to do when an element cannot be used? There must be a strategy to replace the function of that element in the current plan.
- Autonomy is scripted to apply strategies, but it is more powerful when scripted at a high level of abstraction, that is, at a more generic KNOWLEDGE level. Where concepts are used instead of just data and information.

Autonomy Approach

- During a mission, autonomy is achieved with four knowledge domains: System Domain Model (SDM), ISHM Domain Model (IDM), Autonomy Domain Model (ADM), and Mission Planning Domain Model (MPDM).
- ISHM determines health and updates the SDM, Planer determines a course of action, and autonomy strategies help the Planner modify the course of action while considering health and objectives of a mission.

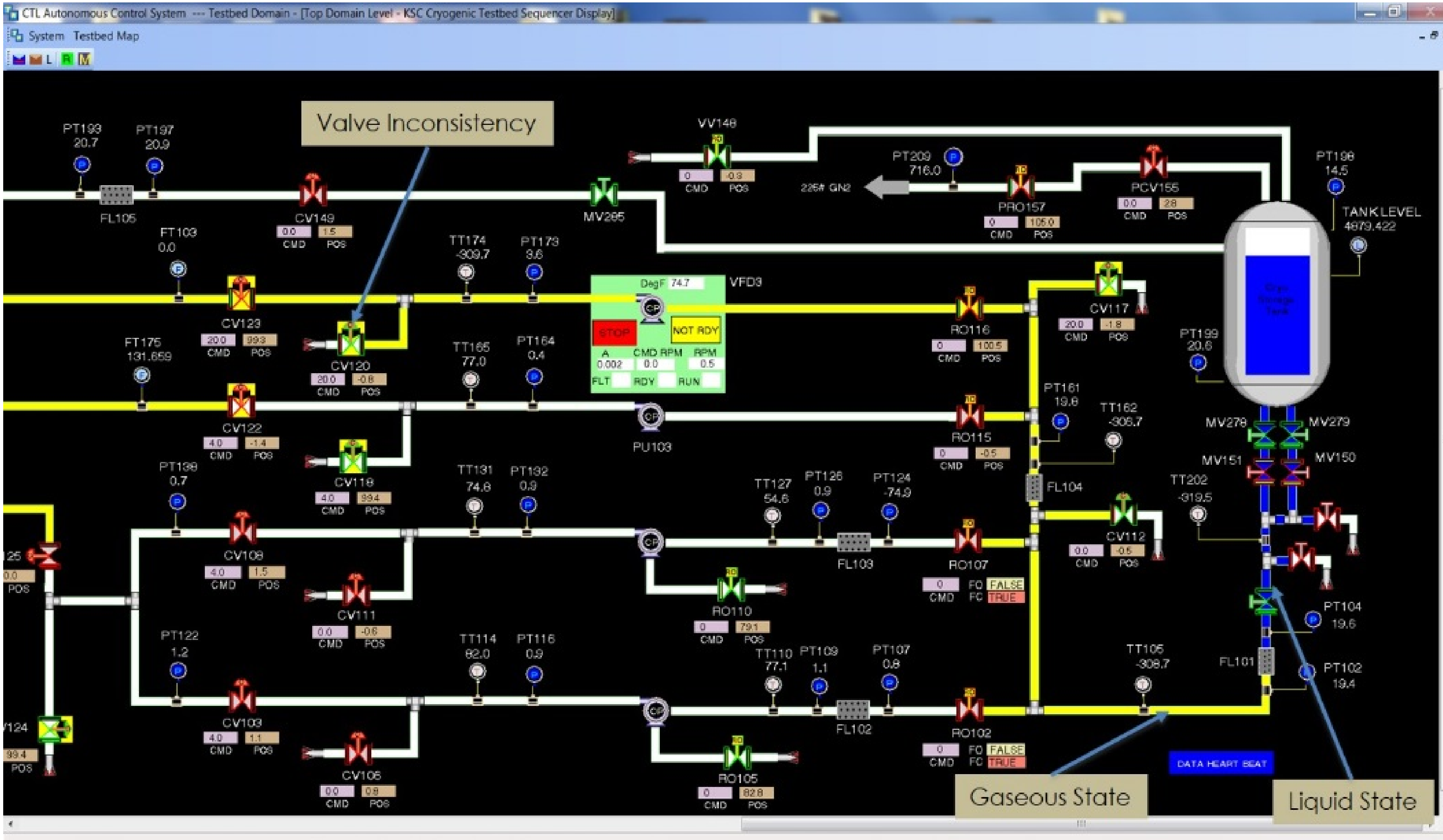
Autonomy Functional Diagram



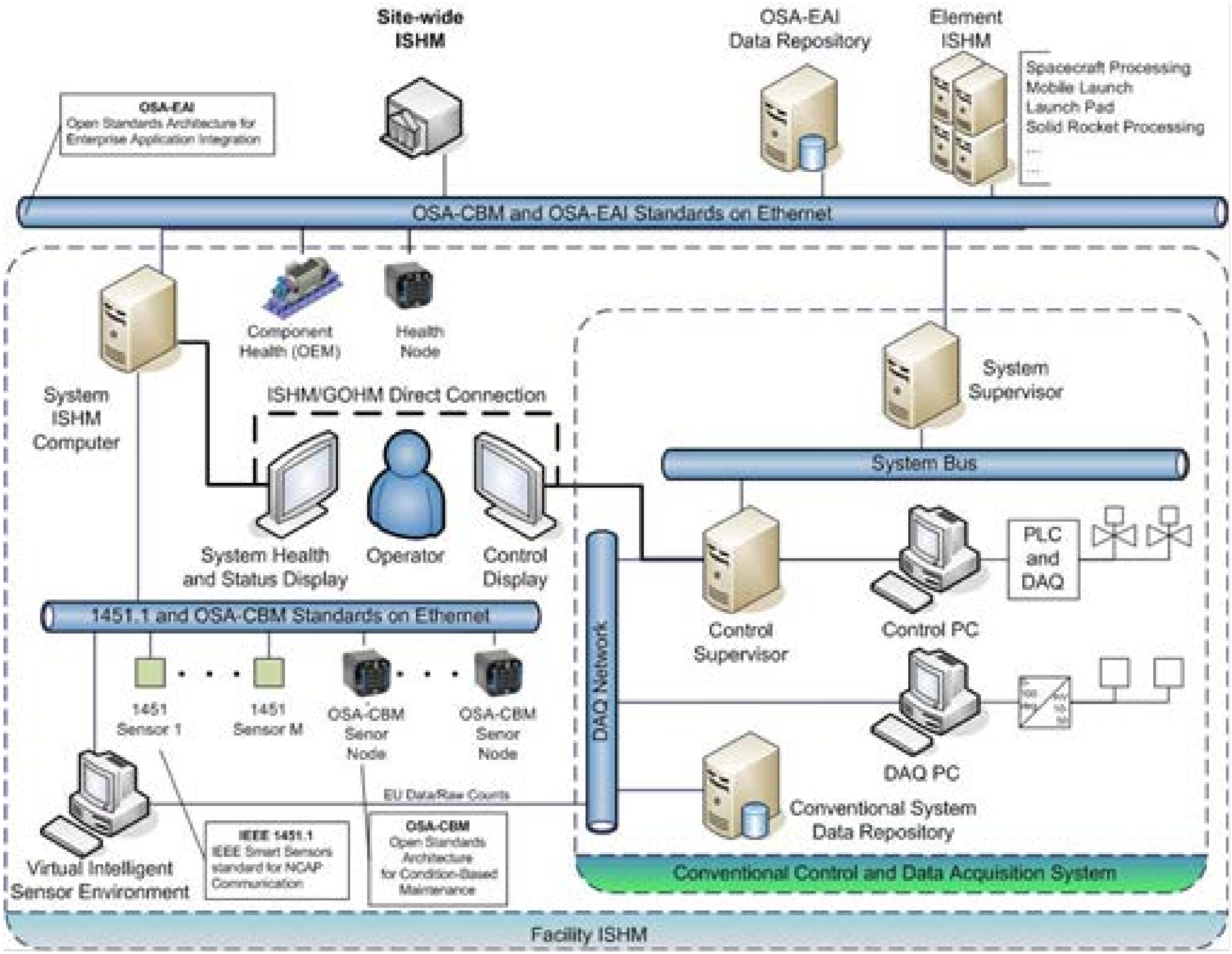
3 Software and Hardware Architectures for ISHM and Autonomy

- The software must enable the creation of domain models of the system which encapsulates information and knowledge about all elements of the system and processes that can take place throughout the system (a knowledge base).
- The software and hardware architectures must make possible data, information and knowledge (DIaK) management, such that data and information is available to any element of the system when needed and for the right context.

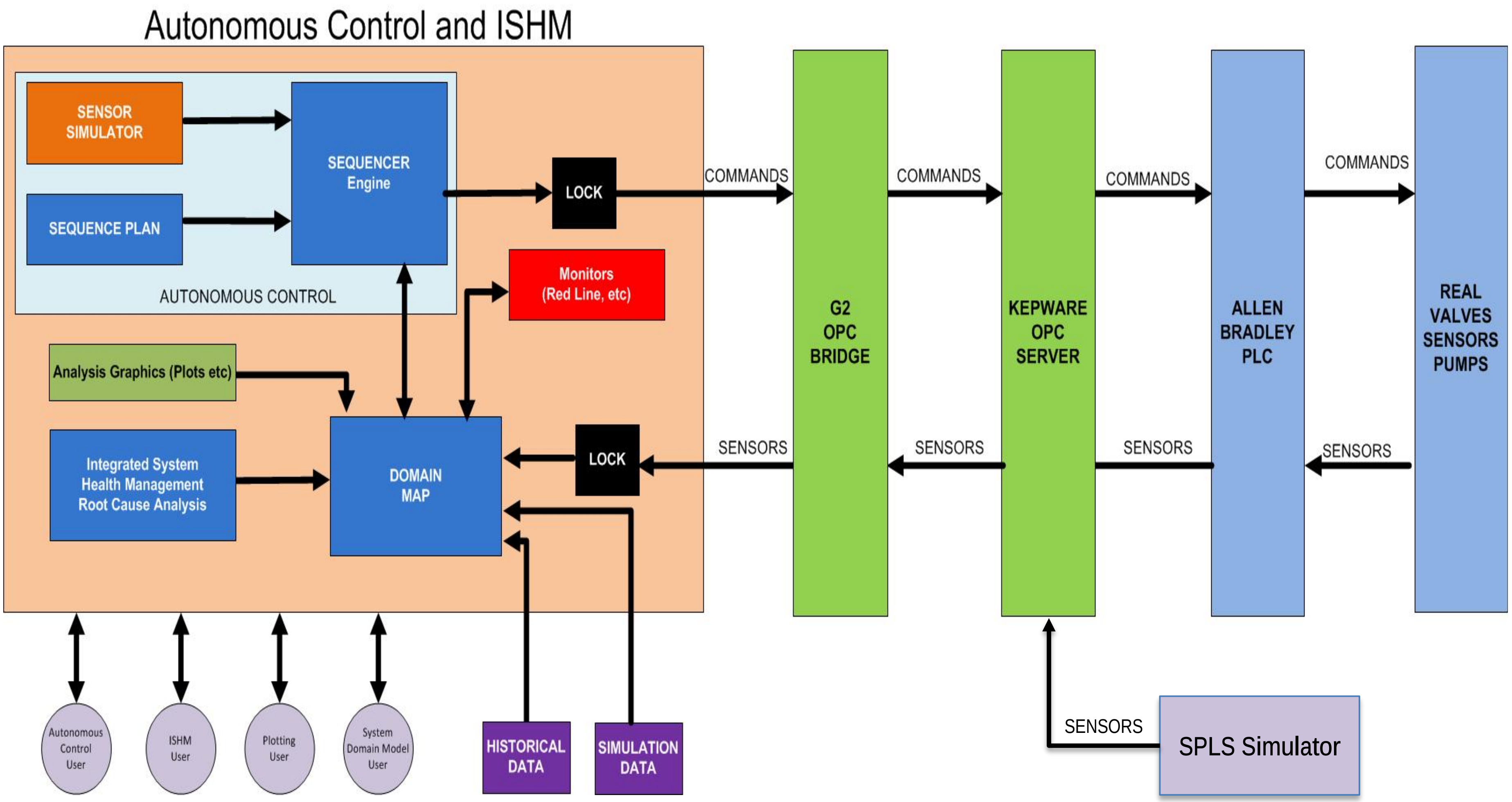
Example Domain Model Representation



Hardware architecture showing for distributed ISHM-AC capability implementation.



AC-ISHM Integration Architecture



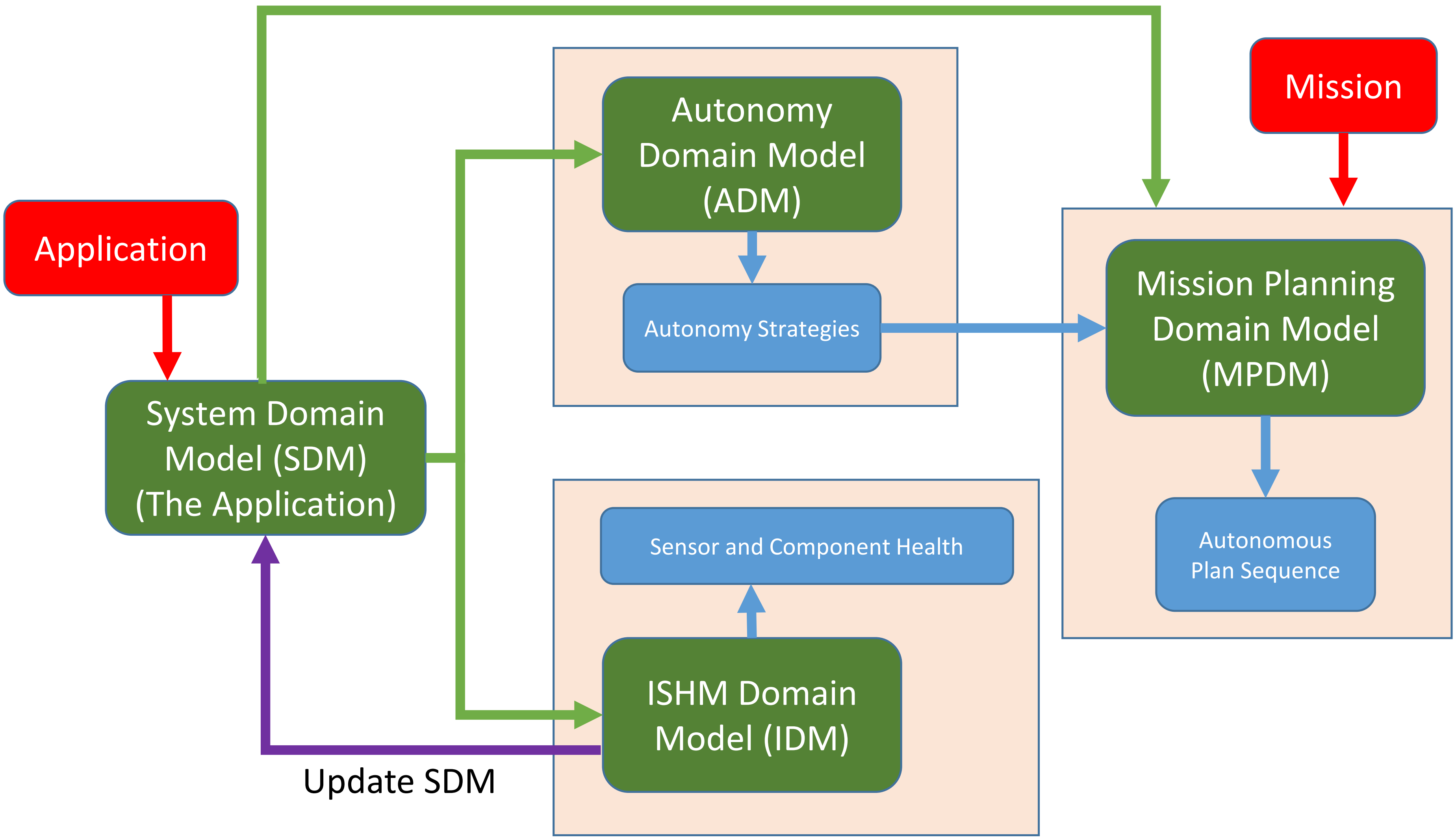
Software Architectures

- **Software architectures for ISHM and Autonomy must enable the implementation of paradigms to employ knowledge and information to achieve the desired functional capabilities.**
- **In the case of ISHM, the goal is to determine the condition of every element in the system, by addressing the following: (1) anomaly detection, (2) diagnostics, (3) prognostics, (4) user interfaces for integrated awareness by the operator.**
- **A software architecture for autonomy should enable the following capabilities:**
  - Mission planning.
  - Resource assessment for mission execution.
  - Strategies to address availability of resources to carry out a mission.
  - Strategies for mission re-planning to deal with unexpected circumstances.

Domain Models and Basic Knowledge Constructs

- **Domain models are needed to encapsulate information and knowledge associated with the system that is to operate autonomously.**
- **Four domain models provide the foundation for autonomy: (1) System Domain Model (SMD), (2) ISHM Domain Model (IDM), (3) Autonomy Domain Model (ADM) and (4) Mission Planning Domain Model (MPDM).**

Autonomy Software Paradigm with Domain Models



Domain Models for Autonomy

- **System Domain Model (SDM)**
  - This is the application domain model.
  - Encapsulates all elements in the system, generally created from design diagrams (e.g. piping and Instrumentation diagrams or P&IDs).
- **ISHM Domain Model (IDM)**
  - Encapsulates knowledge to achieve ISHM functionality (anomaly detection, diagnostics, prognostics, user interfaces for integrated awareness.
  - Uses the SDM and updates ISHM parameters in the IDM to be consistent with the current condition of its elements.

Domain Models for Autonomy (Continued)

- **Autonomy Domain Model (ADM)**
  - Strategies to enable autonomy.
    - Determination of potential replacement elements (e.g. sensors).
    - Determination of alternate flow plaths.
  - Uses DIaK from the SDM and system condition information from the IDM.
- **Mission Planning Domain Model (MPDM)**
  - Creation of mission plans to achieve an objective.
  - Real time modification of mission plans guided by information from the ADM.
  - Uses DIaK from the SDM.

Reusability

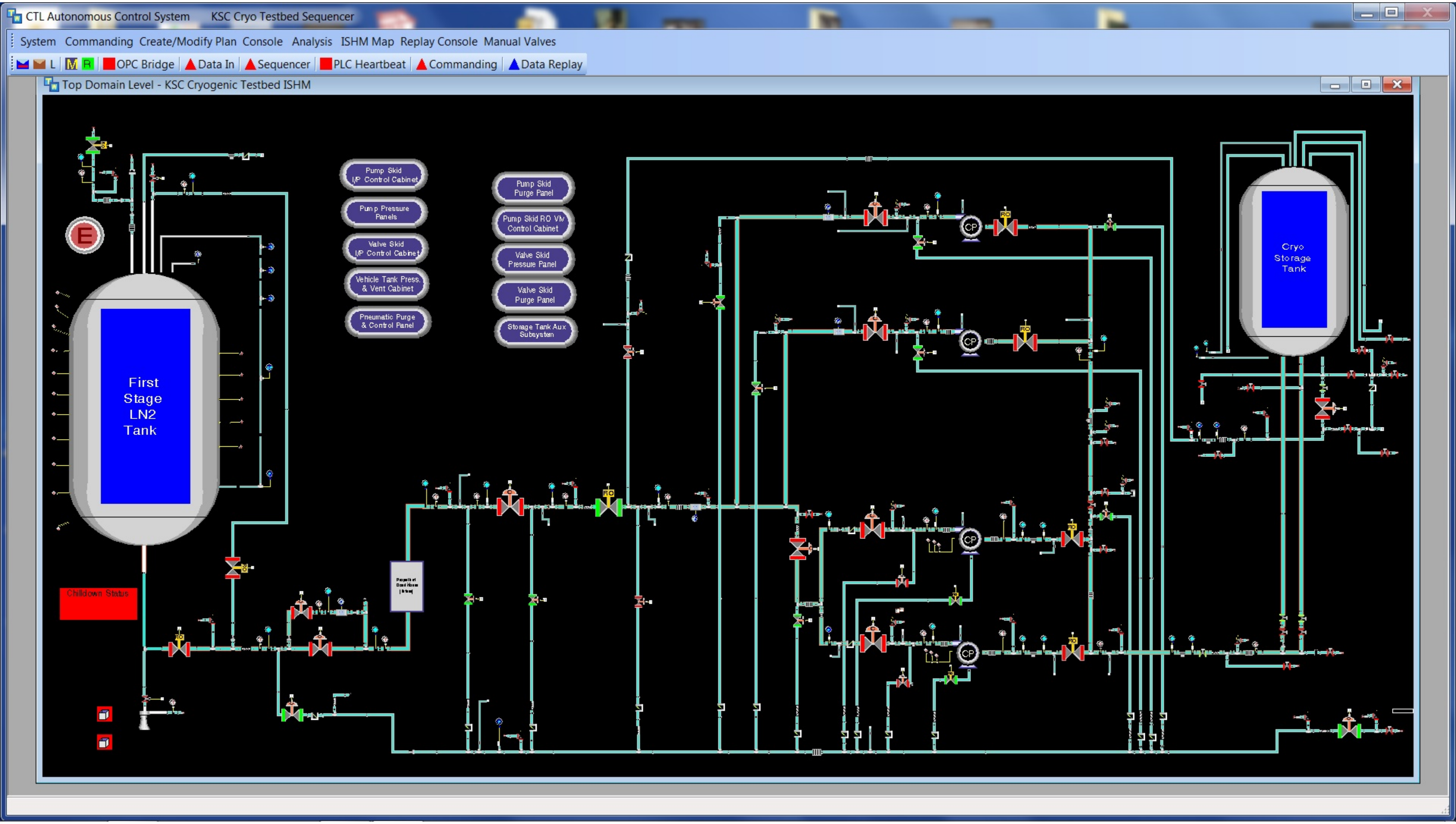
- DIaK to implement autonomy has a substantial portion of re-usable code.
- Strategies for health management and autonomy can be applied to many classes of systems.
- The reason is that these strategies are founded in engineering principles that apply to classes of systems.
- For example a leak detection strategy may be to observe isolated subsystems (subsystems isolated by valves that are closed), and see if pressures are not steady.

Evolution

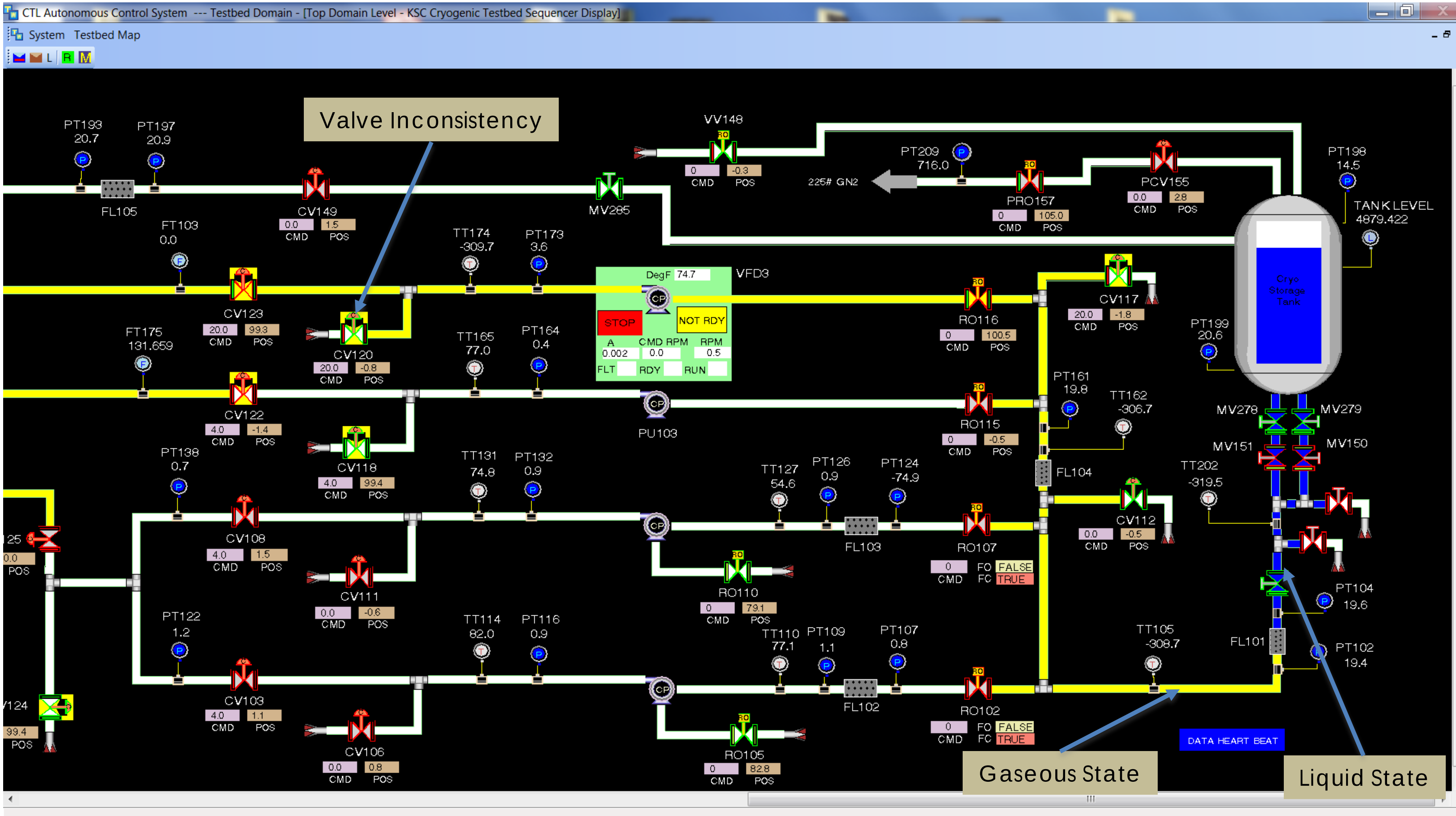
- Autonomous systems are intelligent systems by definition.
- They incorporate information and knowledge that is evolving in time.
- Architectures and software environments for autonomy must enable systematic evolution of the capability.
- for example, any time there is a new strategy to determine that a sensor is faulty, one should be able to implement that strategy and make it part of the existing knowledge base of the domain model in a systematic manner, without affecting existing code. And the new strategy should be integrated automatically.

Pilot Implementation: Autonomous Management of Cryogenic Fluids at a Rocket Launch Pad Testbed

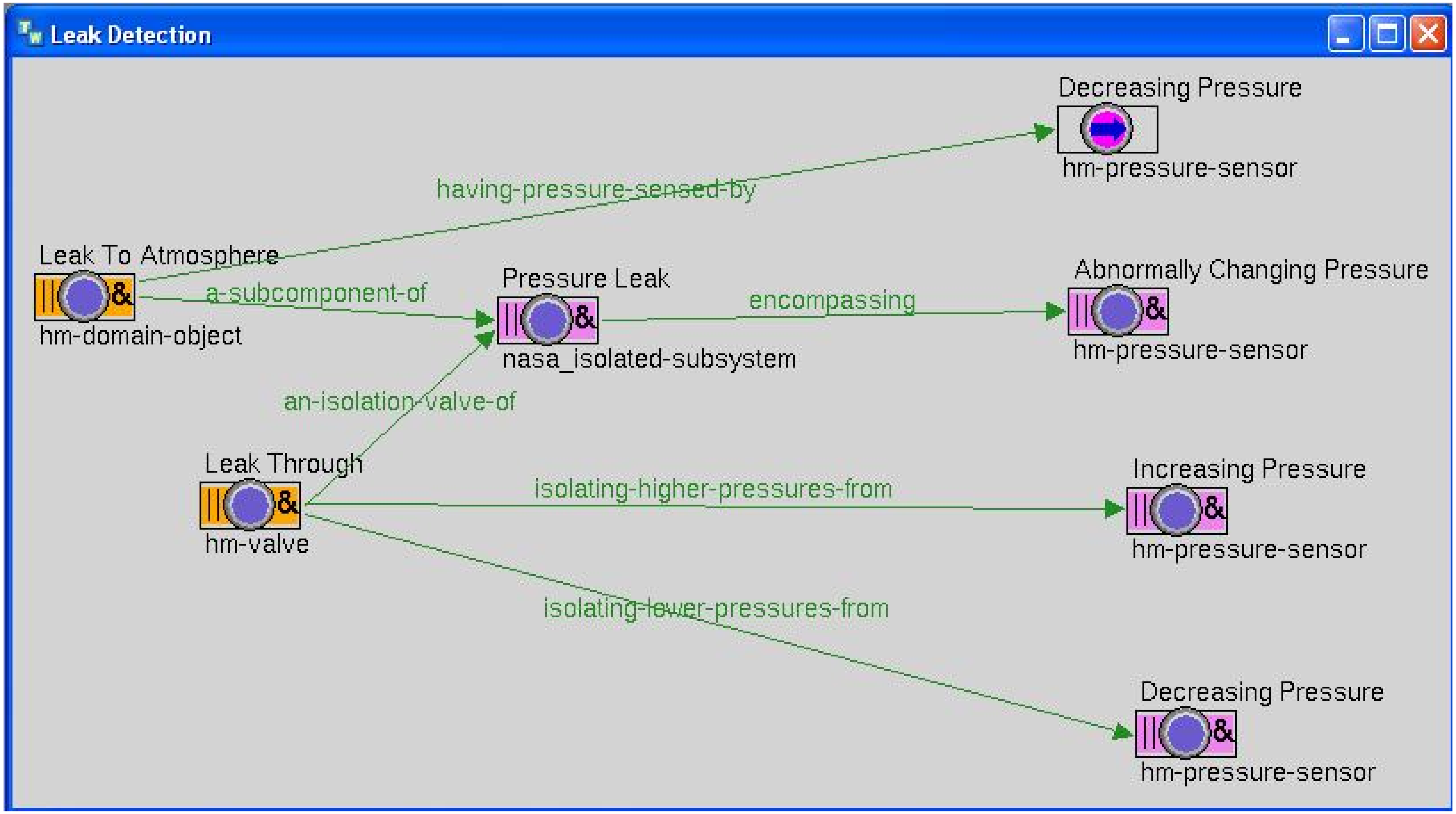
System Domain Model: Top Domain Map



Autonomous Control Operations Client to the Knowledge Domain Model



FMEA Program for Leak Detection



Autonomous Control Sequence Capability

1. Sequence Creation, loading, and execution.
   a. Sequences can be seen as mission plans.
   b. The system enables quick and easy creation of sequences with menus.
   c. Sequences are represented in tabular and graphical formats.
   d. Sequences can be verified by simulating conditions that enable actions to be executed.
   e. Sequences can be saved and loaded as needed.
   f. Sequence conditions include:
      a. Sensor triggers.
      b. Redline triggers.
      c. Fluid saturation state.
      d. Redline sensor failure
      e. May include other conditions from health or other algorithm outcomes.
   g. Sequence actions include:
      a. Valve and pump operations.
      b. Camera pointing.
      c. Execution of special sequences such as shut-down or reverting to a prior step.
      d. May include execution of any sequence that responds to system conditions and planning.

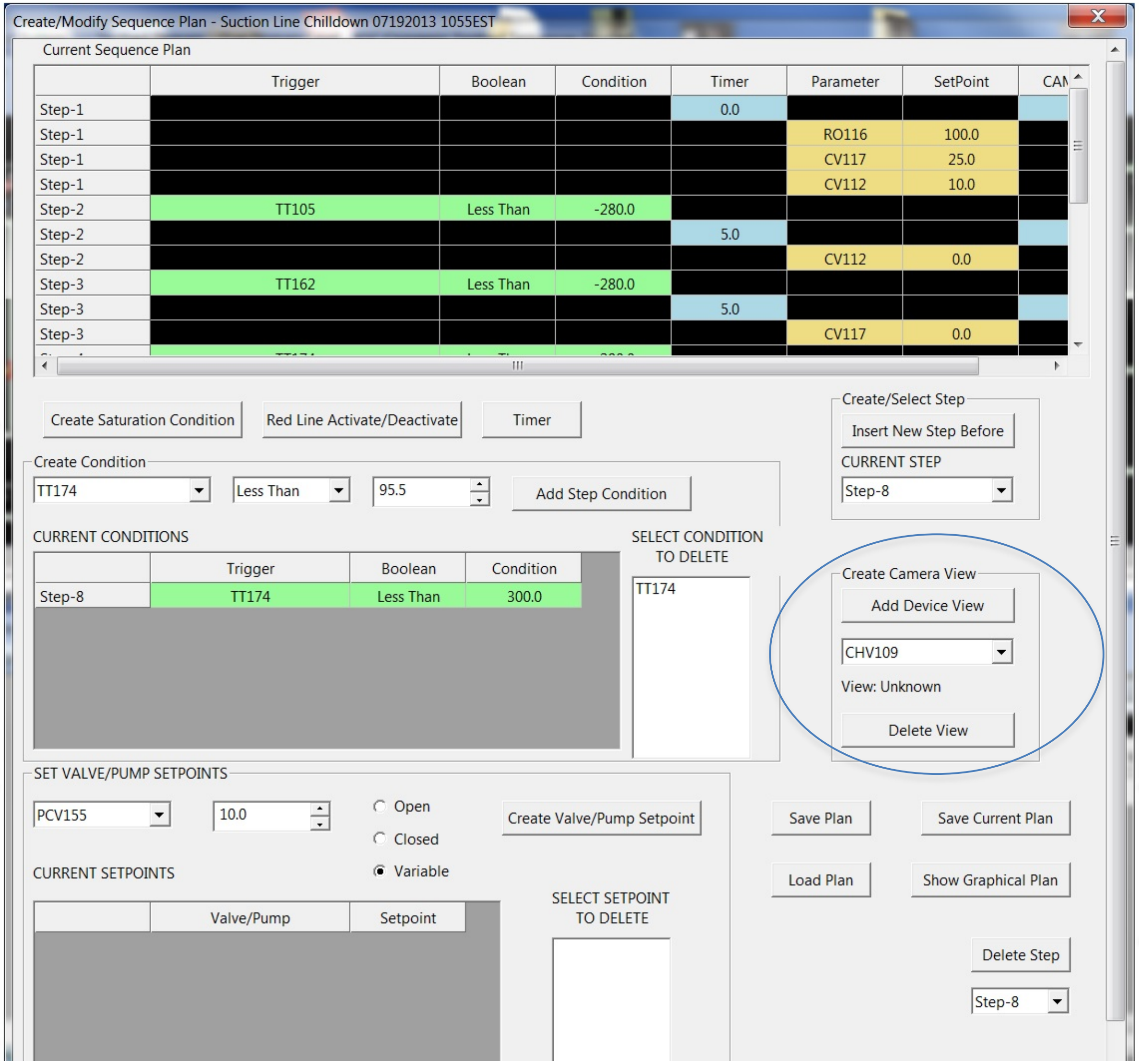
Autonomous Control Sequence Creation and Simulation

Sequence

Define
Saturation
Redlines
Timers

Define
Sensor
Trigger
conditions

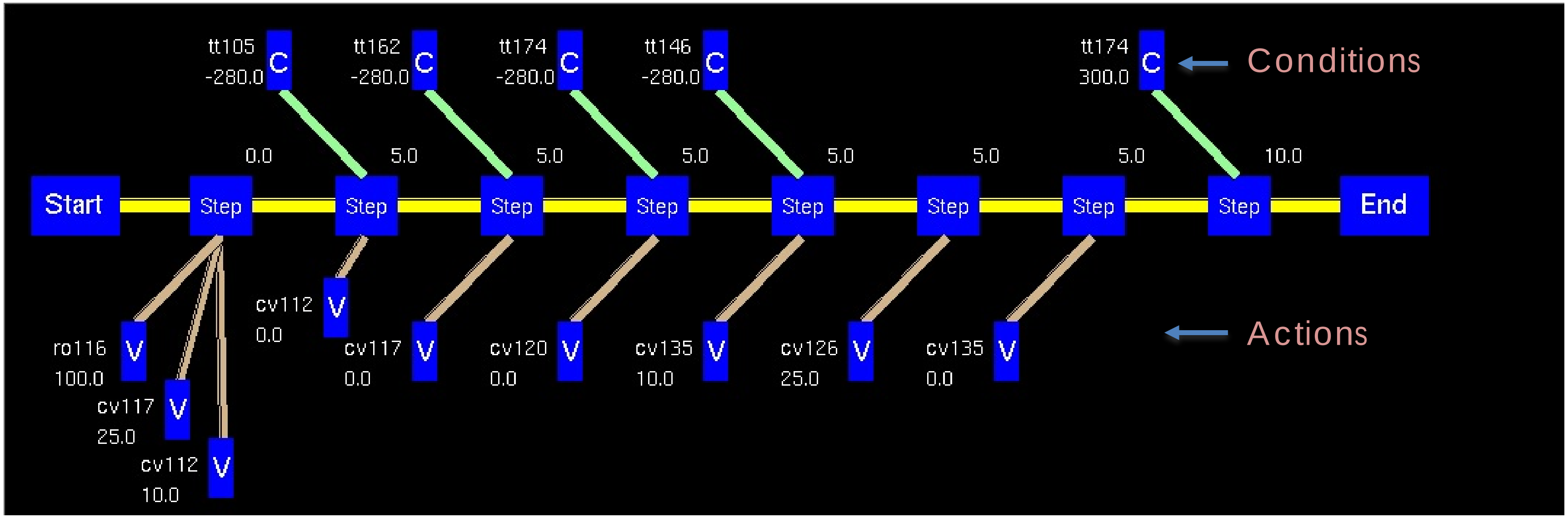
Define
Valve and
Pump
Actions

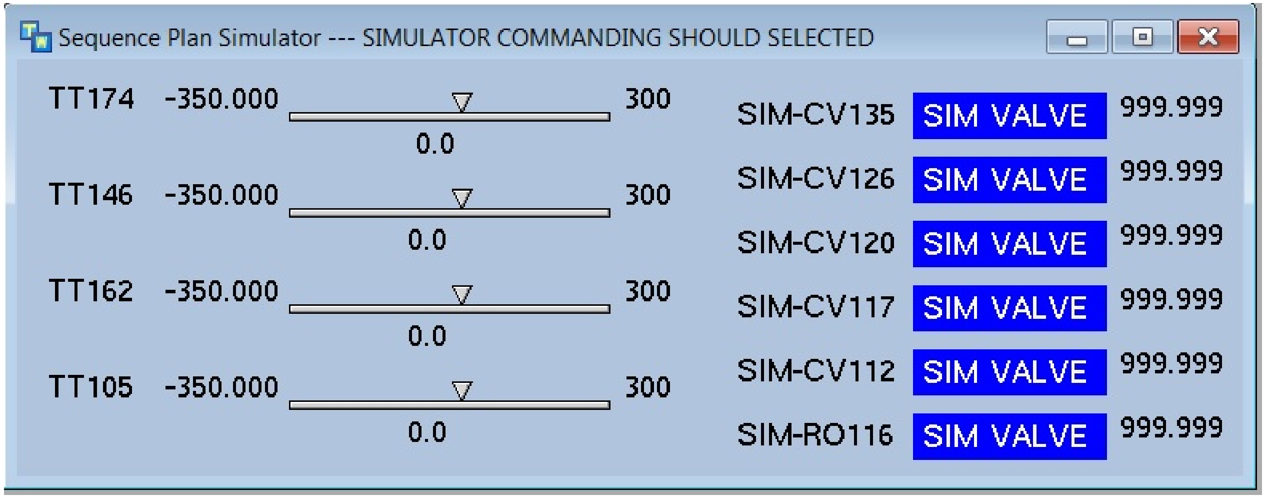


Camera View Definition

Save and Load Sequences

Autonomous Control Sequence Simulation





Autonomous Control Sequence Execution

SEQUENCE PLAN CONTROL - Simple_RL_Nominal_Short_CVCond_wTimer_wPump_v3

STEPS EXECUTED

	Step Label	Trigger	Boolean	Condition	Timer	Valve	Set Point	CAM View
Step-2-3						CV120	75.0	
Step-2-3	1030				13.0			
Step-2-3		TT174	Less Than	-125.0				
Step-2-3		CV112.POSITION	Greater Than	20.0				
Step-2-3		CV117.POSITION	Greater Than	95.0				
Step-2-3		RO116.POSITION	Greater Than	95.0				
Step-2-2						CV117	100.0	
Step-2-2						CV112	25.0	
Step-2-2						RO116	100.0	
Step-2-2						TIMER-1	0.0	
Step-2-2	1020				12.0			
Step-2-2		PT199	Greater Than	9.5				

STATUS: RESET

START | RESET | PAUSE | RESUME | FORCE STEP ADVANCE | SENSOR HEALTH CHECK

ACTIVE STEP

COUNTDOWN TO ACTIVATE STEP: 0 Seconds

	Step Label	Trigger	Boolean	Condition	Timer	Valve	Set Point	CAM View
Step-2-4		CV120.POSITION	Less Than	80.0				
Step-2-4		TT162	Less Than	-280.0				
Step-2-4						LC155	15.0	
Step-2-4						CV112	0.0	
Step-2-4	1040				0.0			

Autonomous Control: Monitors and Console

ALL Monitors - Click STATUS to manually ACTIVATE/DEACTIVATE --- Click PLAN to SELECT PLAN or set to NO PLAN

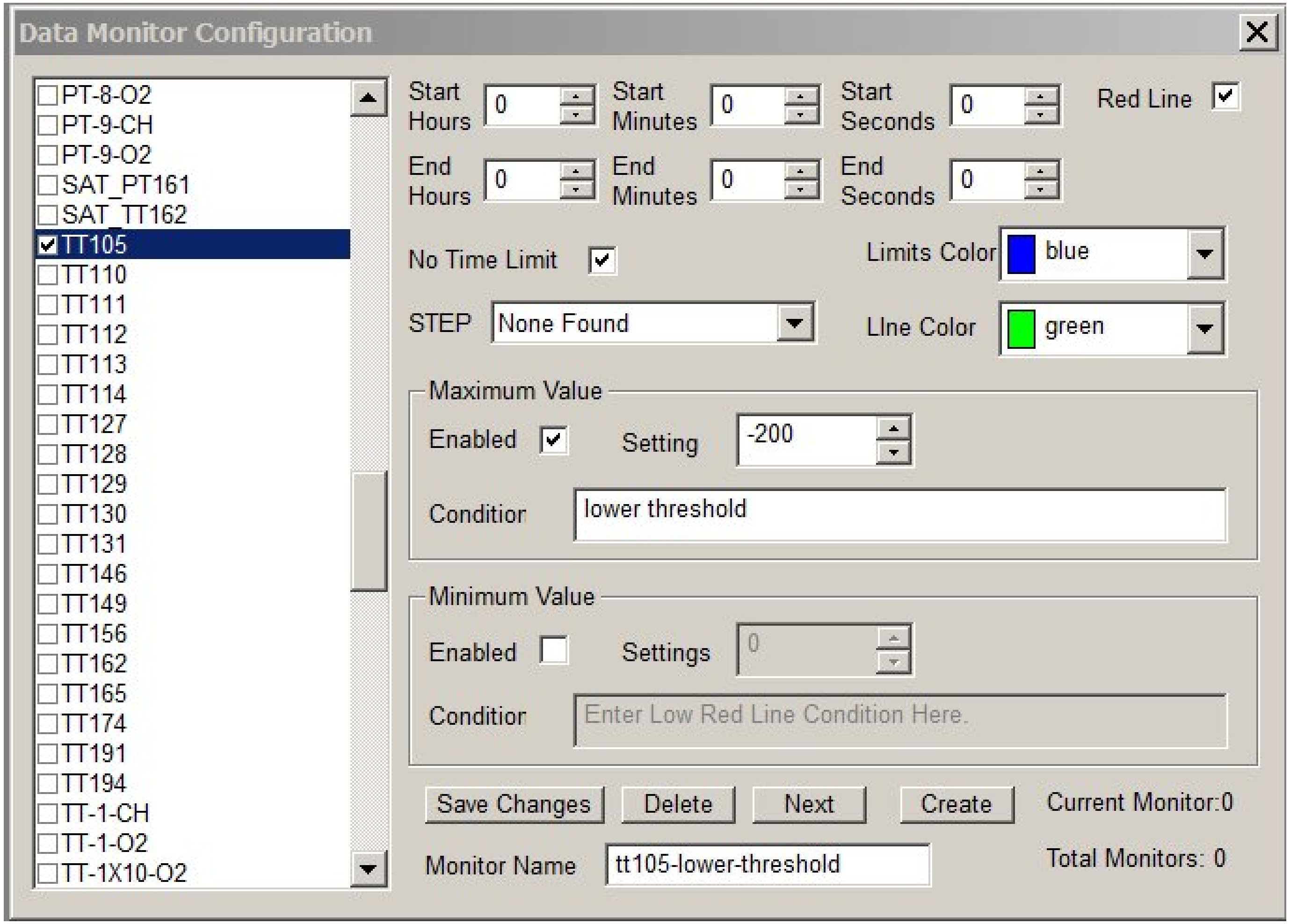
	Status	State	Alternate	Plan	Triggered	Lower Limit Active	Lower Limit	Higher Limit Active	Higher Limit
PT-9-CH	Activated	HEALTHY	PT-10-CH	Adv_to_Shutdown	Not Triggered	Yes	-100.0	No	0.0
TT105	Triggered	HEALTHY	UNKNOWN	emergency1	5/20/15 11:38:26 a.m.	No	0.0	Yes	-200.0
PT-10-O2	Deactivate	HEALTHY	UNKNOWN	multiple-test-plan-2	Not Triggered	No	0.0	Yes	25.0

Console

Options | Delete Msgs | Active | Log

Time	Message
5/20/15 11:39:11 ...	Logging to c:\ishm-nasa-toolkit\log\5M-20D-11-39-log.txt
5/20/15 11:38:26 ...	All sequence plan sensors are healthy
5/20/15 11:38:26 ...	All sequence plan sensors are healthy
5/20/15 11:38:26 ...	RED Line Monitor tt105-lower-threshold triggered by TT105 has exceeded high thres

Autonomous Control: Monitor Creation



On Going Development

- Will use validated technology to implement autonomous propellant loading on a pilot launch system at KSC.
- Target mobile launch class systems.
- Will demonstrate autonomous operations of multi-tank systems, managing simultaneously operational sequences for cryogenic oxidizer and fuel systems.

Backup

CTL System Description

- **HM-PIPE: 2168 (Pipe elements)**
- **HM-PRESSURE-SENSOR: 62 (Pressure Sensors)**
- **HM-TEMPERATURE-SENSOR: 22 (Temperature Sensors)**
- **HM-VALVE: 286 (Valves)**
- **HM-ELECTRICAL-CONNECTION: 100 (Electrical Connections)**