Enabling Autonomous Propellant Loading

Providing Situational Awareness through Model based Reasoning

Mark Walker D2K Technologies
William E. Walker D2K Technologies
Fernando Figueroa PhD. NASA SSC

Joint Conference:
MFPT 2016 and ISA's 62nd International Instrumentation Symposium
May 24-26 2016 Dayton Convention Center, Dayton, OH

Agenda

- APL Overview
  - Objectives
  - Autonomous Operations (AO)
  - Architecture
- AO-MDS
  - Overview
  - Architecture
  - Benefits
- Results and Future Work

Autonomous Propellant Loading (APL)

National Aeronautics and Space Administration

- Part of NASA's Advanced Exploration Systems (AES) mission.
- **Develop and demonstrate systems for autonomous control of cryogenic propellant loading processes**
- Demonstrate certification of Class B, safety critical autonomous propellant loading software for monitoring and controlling UPSS LOX and LCH4 ground systems and SCV Customer's Rocket Cryogenic Propulsion System
- Certified Cryogenic Loading Facility for SCV Activities using LO2/LCH4
  - Capable of supporting customer rocket configuration testing
  - Excellent training facility for commercial and government operator training (skills and processes maturation)

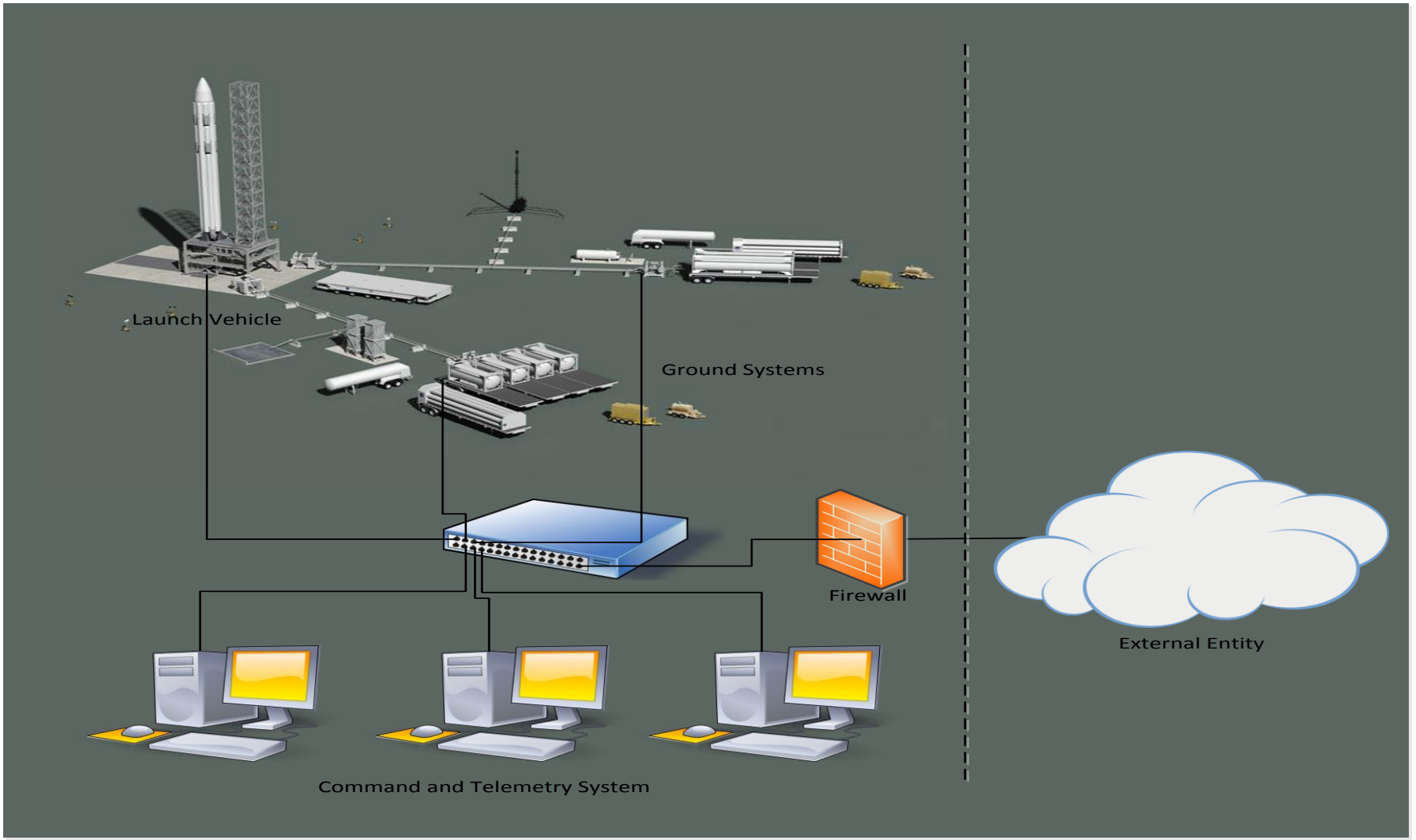
APL Project Milestones

National Aeronautics and Space Administration

- FY15 (July), Demonstrate Multi-stage Autonomous Propellant Loading Using Simulation
- FY16 (July), Demonstrate Multi-stage Autonomous Propellant Parallel Loading using LN2
- FY17 (March), Demonstrate Multi-stage Autonomous Propellant Parallel Loading using LCH4 and LO2 Commodity

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

APL Conceptual Diagram



APL Test Facility

National Aeronautics and Space Administration

Science & Technology Building

Autonomy

- **Autonomy:** the ability for a system to apply self-directed intelligence and adaptation in order to produce a successful response to unanticipated situations.
- There are degrees of autonomy, ranging from low levels to high levels.
- 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.

Strategies for Autonomy

- **Strategies for autonomy guide the decision making process.**
  - Ex: What to do when an element cannot be used? There must be a strategy to replace the function of that element in the current mission 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.

Autonomous Control

- 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.

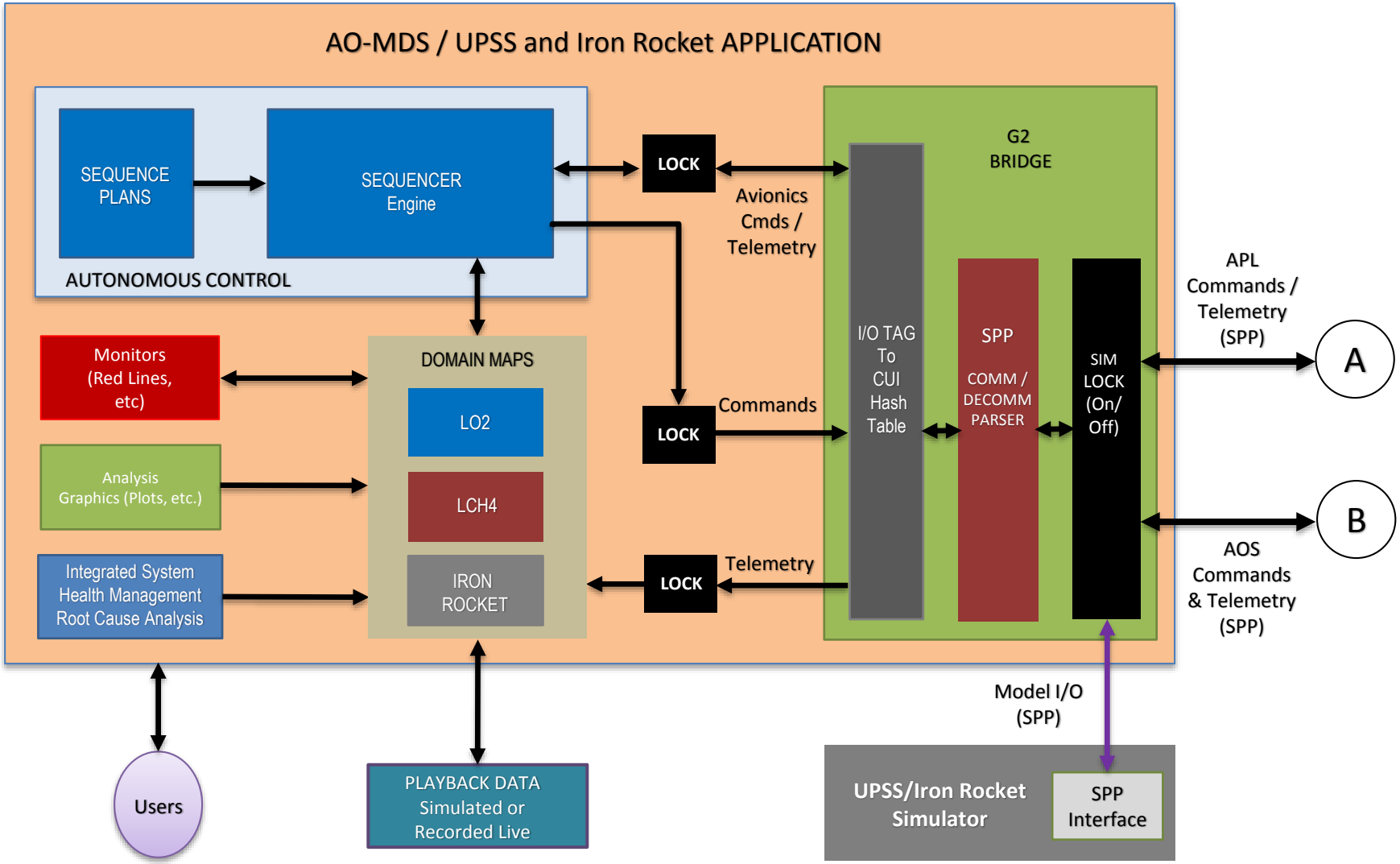
APL Fault Detection

National Aeronautics and Space Administration

3 Parts to APL Fault Detection:

- **Known Failure Case**: Failures Identified by FMEA including critical failures and associated mitigations and instrumentation only failures
- **Potential Failure Case**: Sensor redlines based on engineering judgment
  - Exceeding sensor redlines will terminate loading sequence and initiate drain, team will evaluate data post drain and determine if redline value is valid and if failure occurred – potential to update FMEA if team determines new failure mode not previously identified
- **Less Likely Failure Case**: Trending of sensors data using loading data and pattern recognition (Artificial Neural Net - ANN) to identify when sensor data is out of nominal but not exceeding redline limits

AO-MDS/SIM Software Architecture



AO-MDS/SIM Software Architecture-Description

National Aeronautics and Space Administration

The APL AO-MDS system includes the following modules and interactions.

Autonomous Control: It includes the Sequencer Engine and a Sequence Plan sub-modules. This module enables creation, validation (by simulation of values), and execution of sequences. The module can be locked to stop any commanding to real hardware and allow usage of simulated and/or playback data.

Domain Map Module: Is where all elements of the UPSS system are represented as object instances. It is created from schematics, and configures according to schematics and processes of the UPSS. The representation is graphical (e.g. piping diagrams) but every icon represents an object with a wealth of data, information, and knowledge (DIaK) that is used for inferencing and decision making. Real-time data from the UPSS is streamed using an SPP Bridge for real-time operations. The real-time data can be locked in order to stream data that may come from files. This non-real-time data may be used for replay of actual tests. Simulation data can also be streamed.

Analysis Graphics Module: Provides plotting capability for data.

Integrated System Health Management - Root Cause Analysis: Includes capability for ISHM, such as anomaly detection, diagnosis, prognosis, reporting of anomalies, etc. Any number of users can Access any module of the ISHM-AC System from any computer that is in the network.

MPCDU: Mobile Power Command and Data Unit.

G2 SPP Bridge: A G2 Space Packet Protocol (SPP) UDP network interface for ground and vehicle (Iron Rocket) side I/O.

CCSDS/SPP to CIP Gateway: A 2 -way interface used to convert between SPP and ground side CIP.

Allen Bradley PLC: Programmable Logic Controllers that operate the UPSS ground systems.

Simulator (SIM) Lock: Controls whether G2 PLC tag values (and possible avionics commands) are supplied by Simulator or direct I/O with SPP/CIP PLC bridge/Avionics Interface.

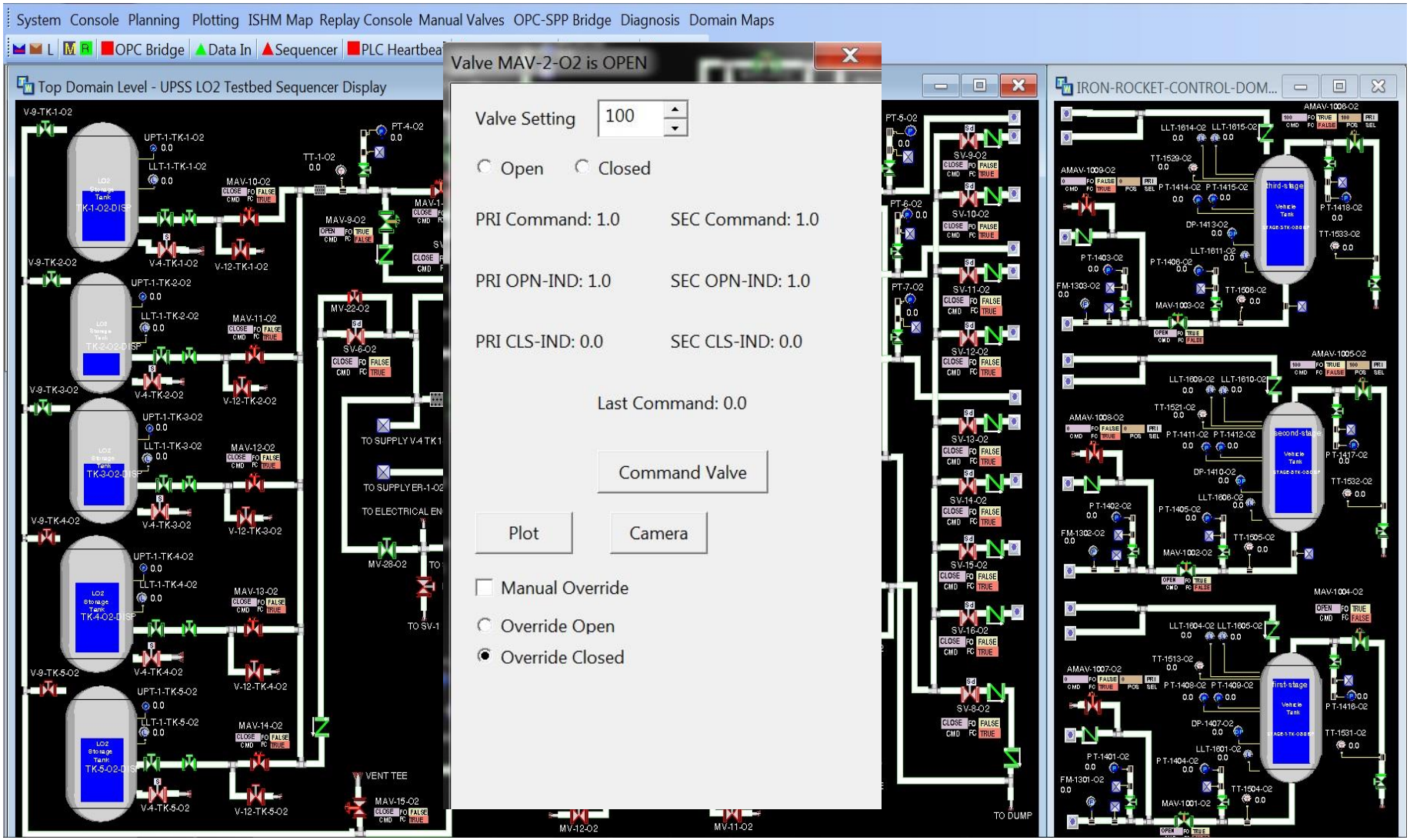
LabViews Tag Simulator: A G2 development and test simulator used to set ground PLC tag values via an OPC server connected to PLCs.

Software and Hardware Architectures

- The software must enable the creation of a domain model 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).

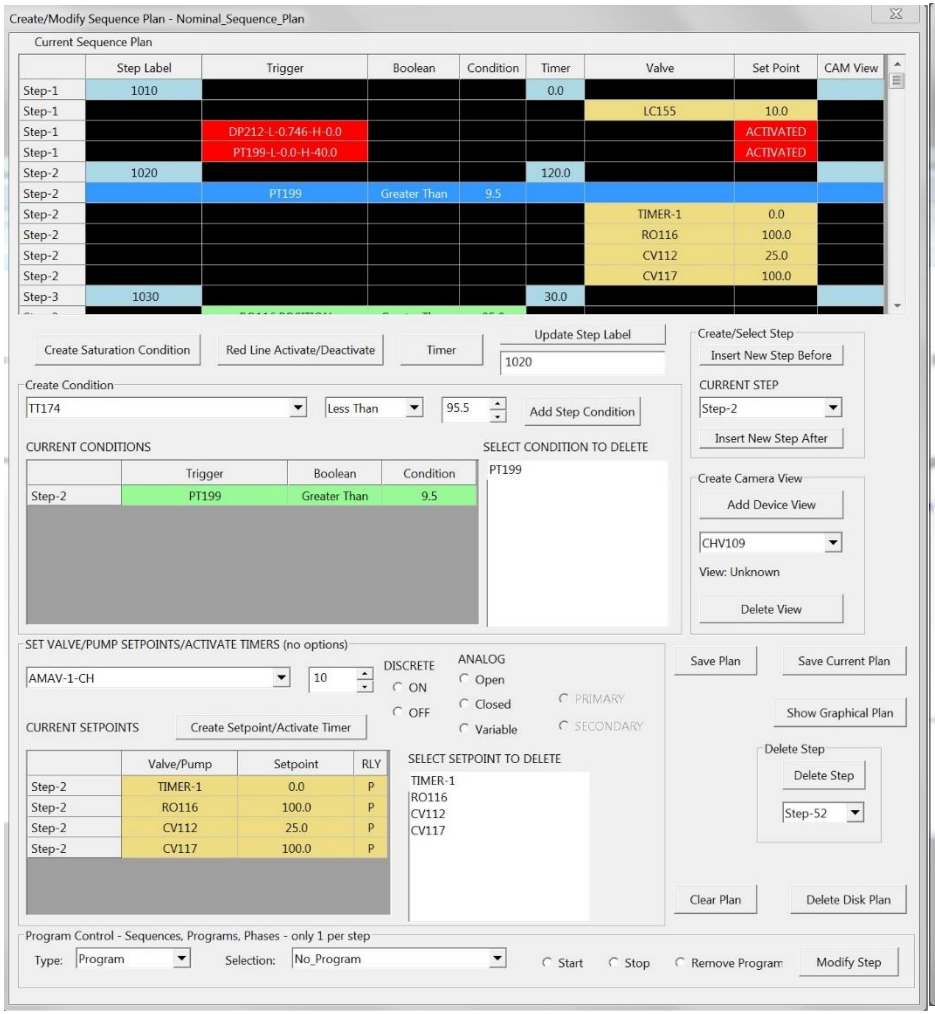
- The software and hardware architectures must perform data, information and knowledge (DIaK) management, such that data and information is available to any element of the system when needed and for the right context.

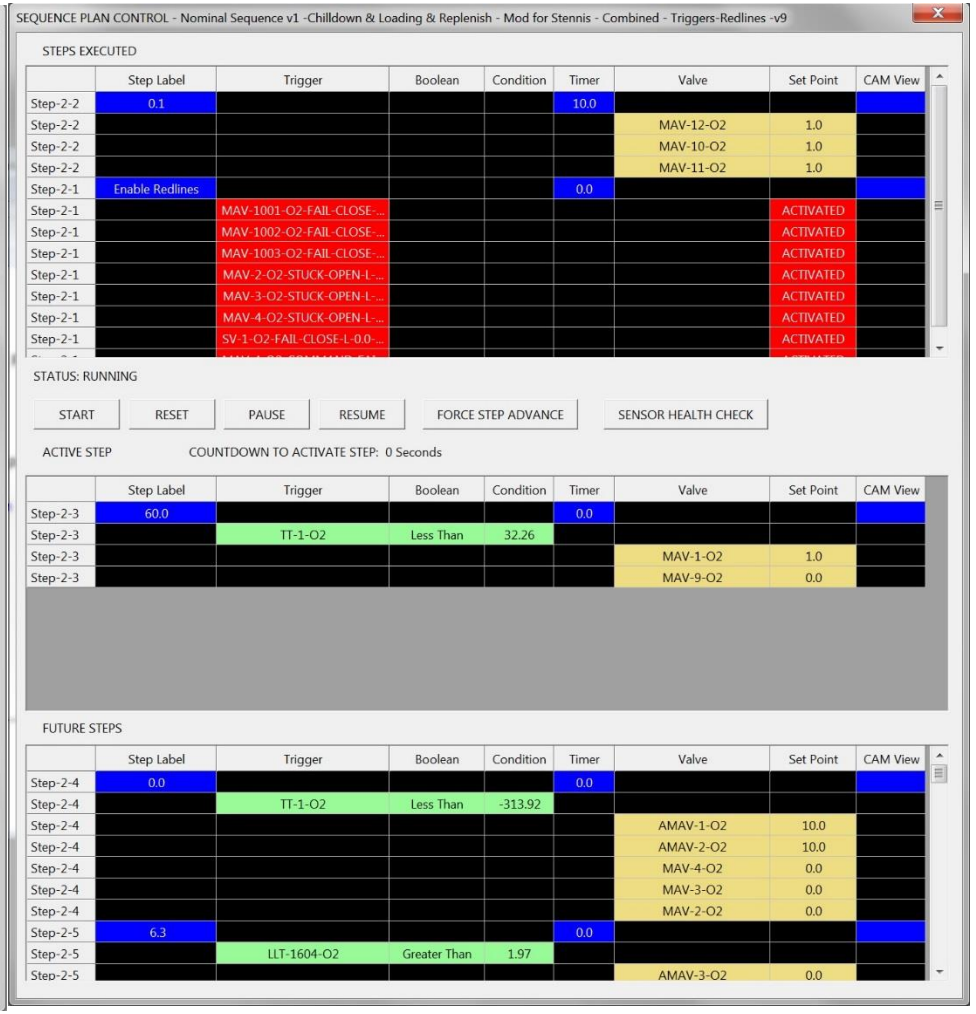
APL Control Domain Representation



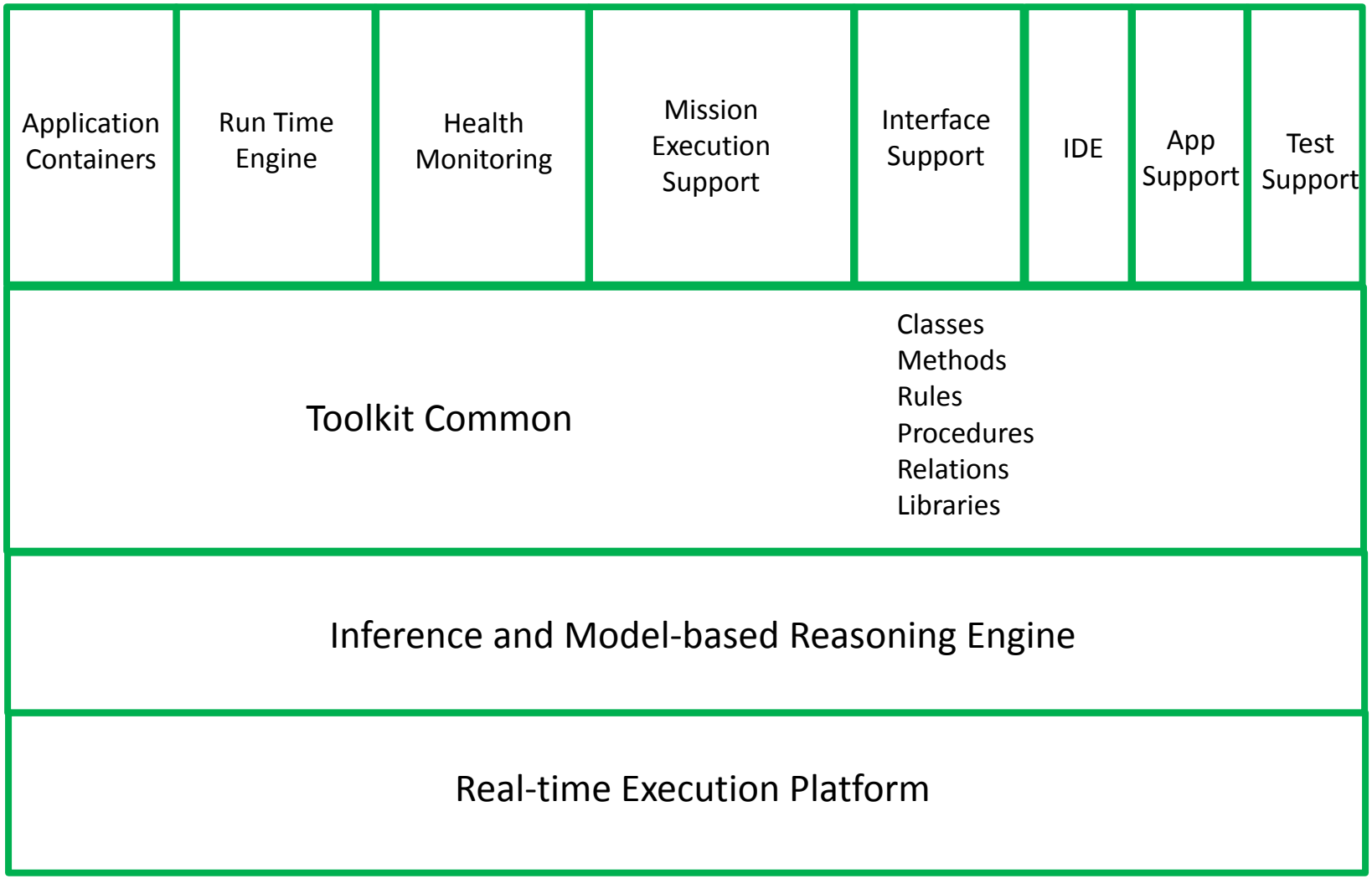
Autonomous Mission Task Execution

National Aeronautics and Space Administration

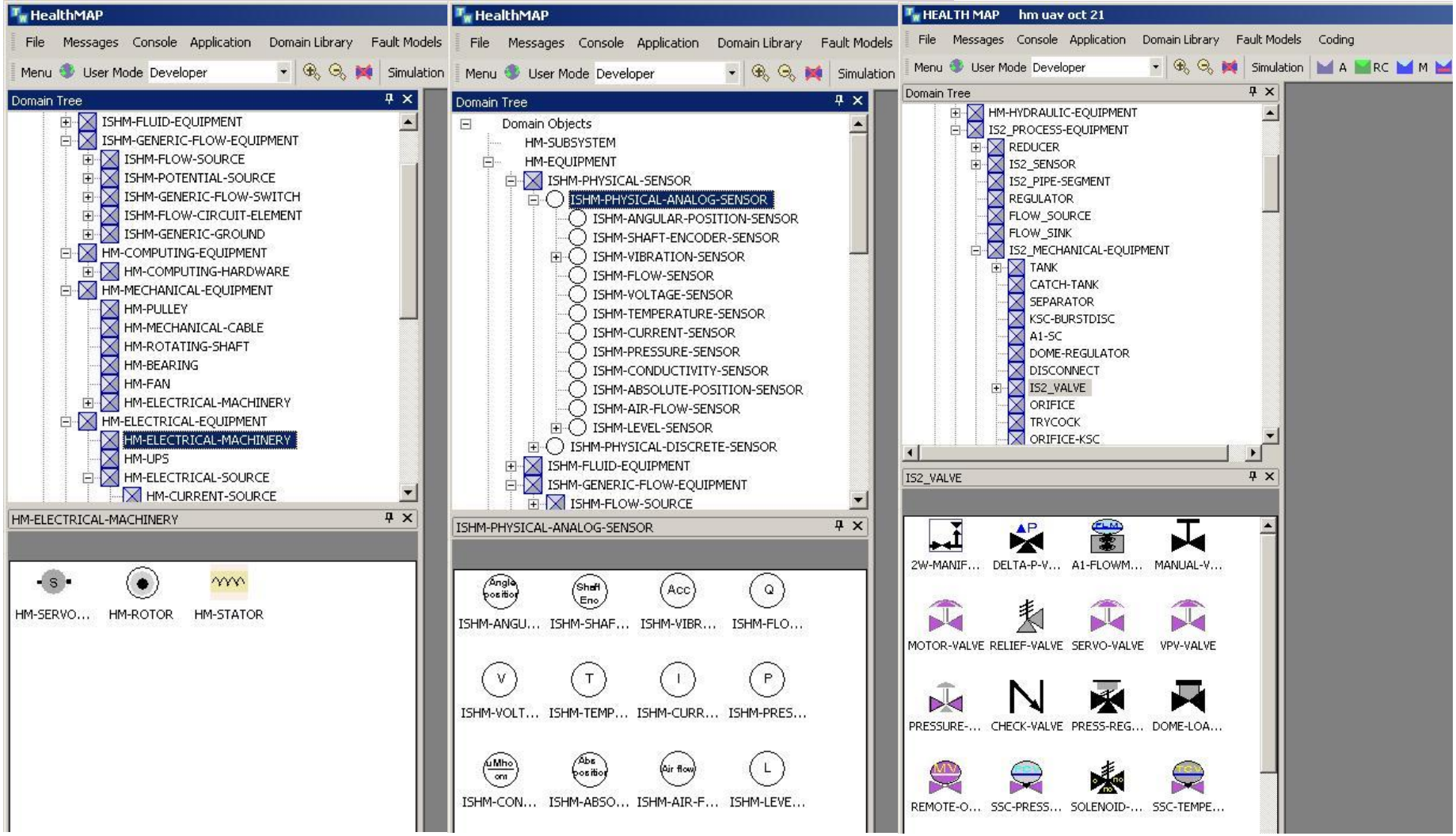




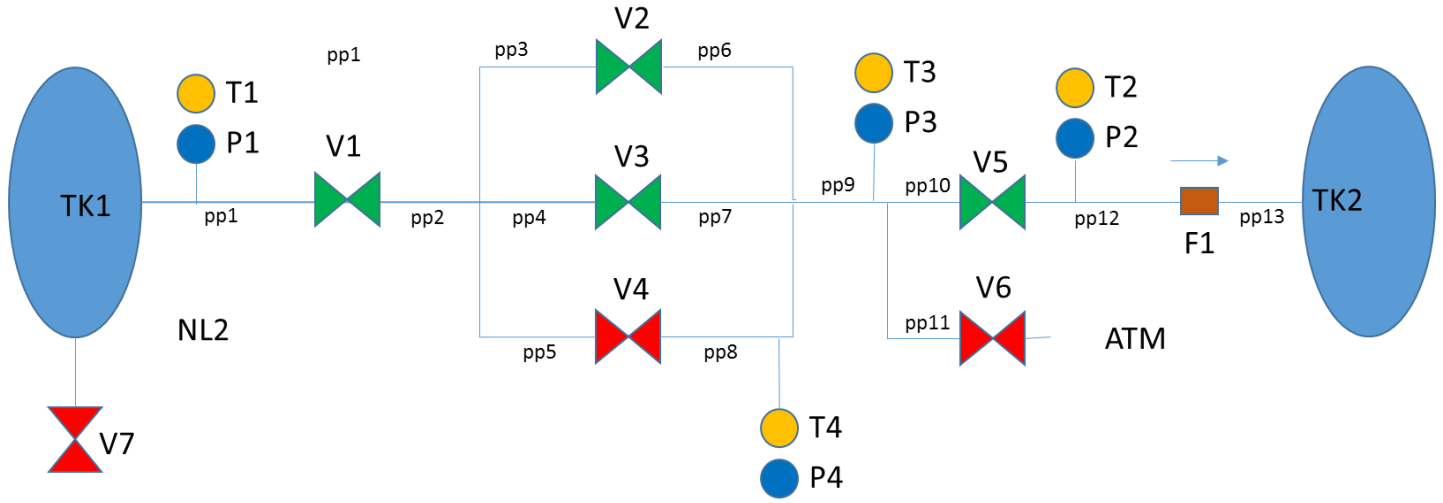
High Level AO-MDS Architecture



AO-MDS: Extensible Model Libraries



Concepts and Models



Flow Subsystem as a Concept

Flow Subsystem 1: Members (TK1, pp1, T1, P1, pp2, pp3, V2, pp6, pp9, T3, P3, V5, T2, P2, F1, TK2), Source: TK1, Sink: TK2.
Flow Subsystem 1: Members (TK1, pp1, T1, P1, pp2, pp4, V3, pp7, pp9, T3, P3, V5, T2, P2, F1, TK2), Source: TK1, Sink: TK2.

Note: AO-MDS incorporates the concept of Flow Subsystem and dynamically determines Flow Subsystems for any application and its current configuration.

In Contrast with a data/information driven approach:

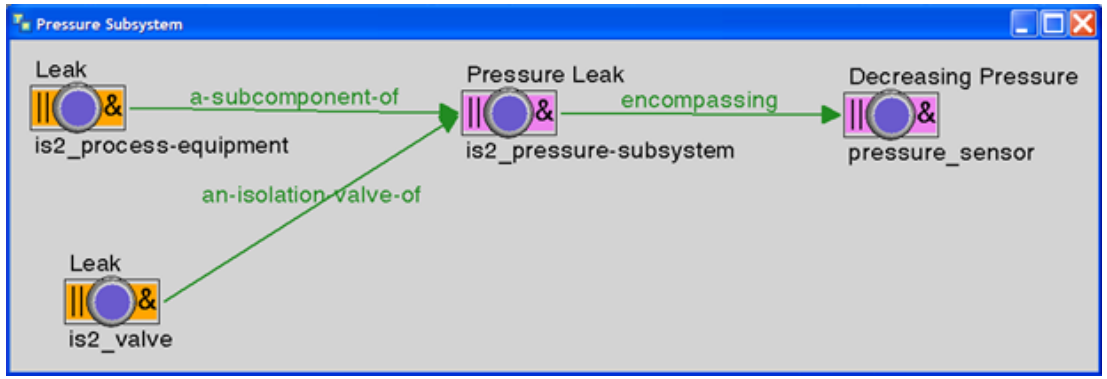
Flow subsystem selected from a pre-defined list that considers all possible combinations of valve configurations for all schematics

- Generally hundreds or thousands of valves are involved, becoming a complex combinatorial problem.
- Any changes in the system (e.g. adding a valve) will require extensive work to update the combinatorial list.
- Any new system will require its own combinatorial list.

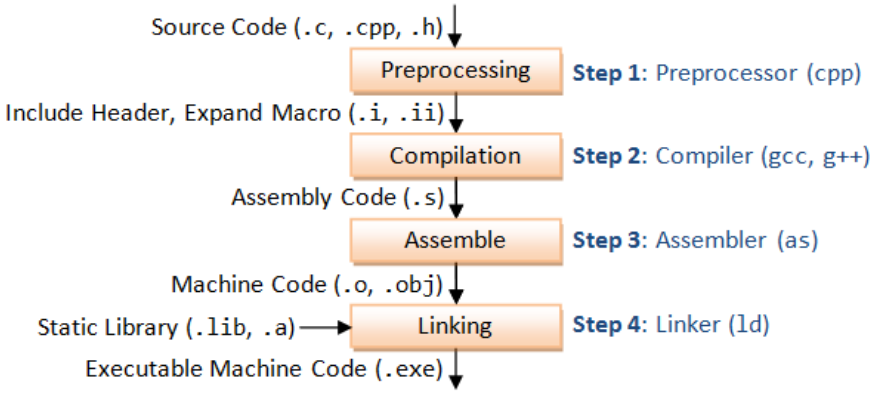
Failures Modes and Effects Analysis (FMEA) Modeling based on MIL-STD-1629A(2)

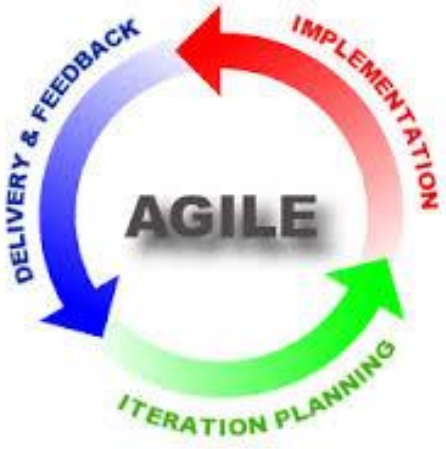
National Aeronautics and Space Administration

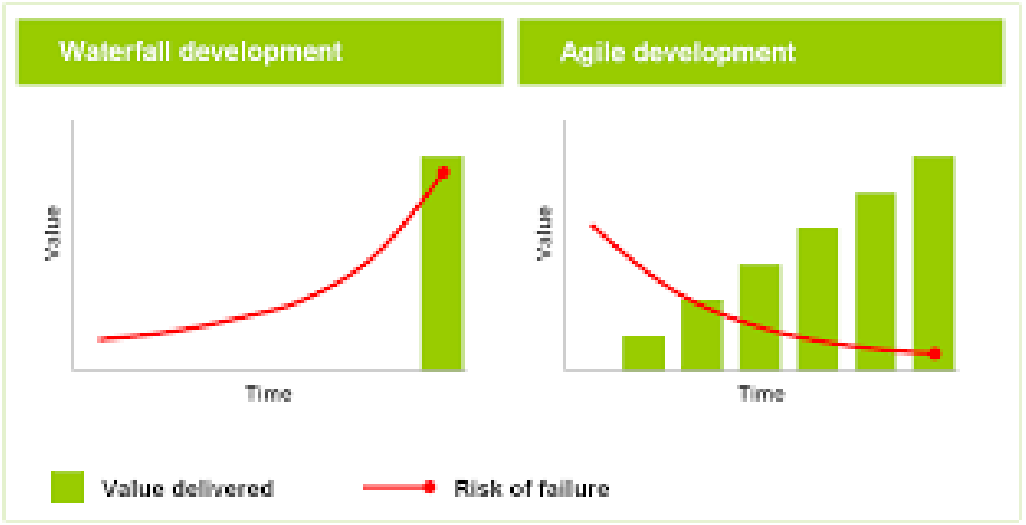
ID #	Item-Functional Identification	Function	Failure Modes and Causes	Mission Phase-Operational Mode	Failure Effects			Failure Detection Method
					Local Effects	Next Higher Level	End Effects	
	Process Equipment	Fluid feed subsystem	Leak	Sealed subsystem maintaining pressure		Pressure leak	Decreasing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



Agile Development Process



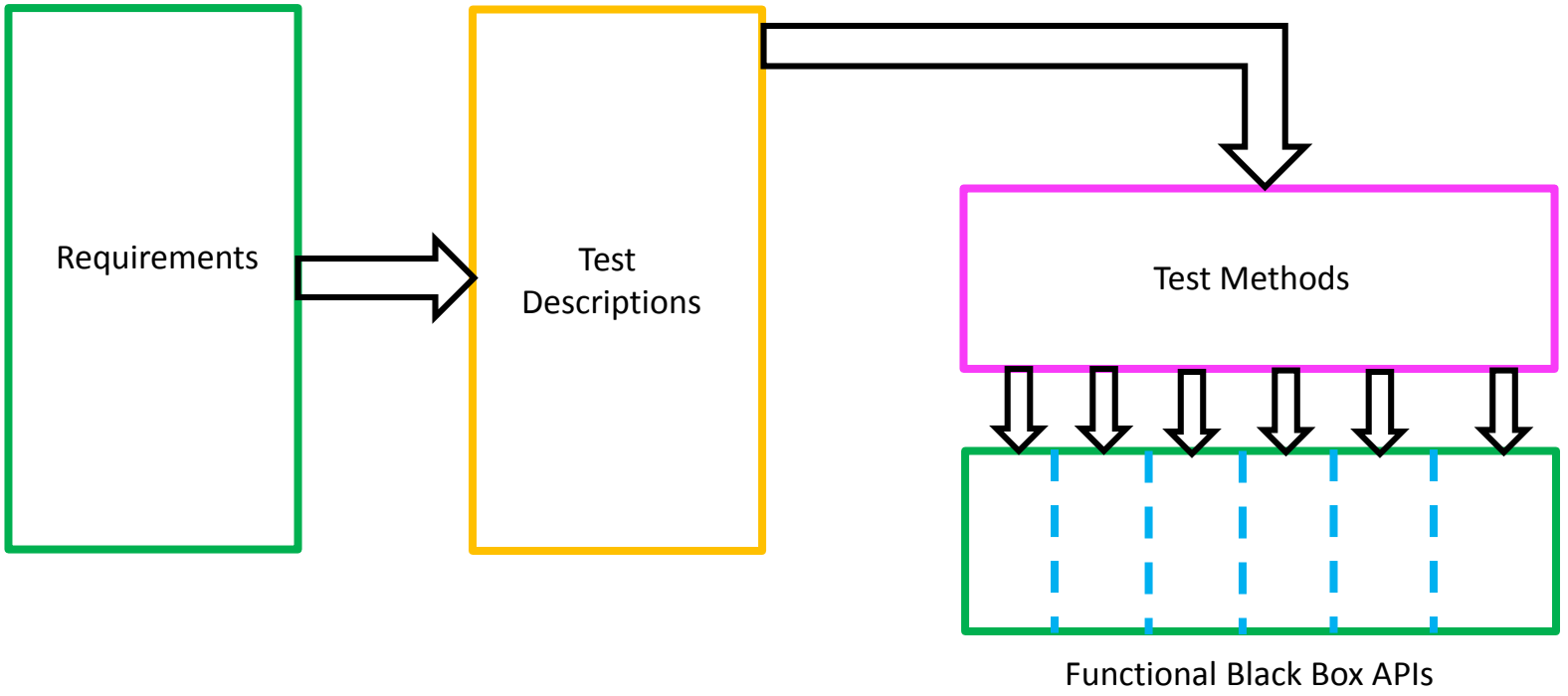




Quest for Software Quality

- Test Driven Design (TDD)
  - Write a test that fails
  - Code until it passes
  - Refactor (re-coding if it breaks)
- Behavior Driven Design (BDD)
  - “BDD is about implementing an application by describing its behavior from the perspective of its stakeholders”
  - Requirements as User Stories
  - Pull vs. Push based
- Automated Testing using philosophy of jUnit, TestNG (example tools)
  - Automated Report Generation
  - Tests follow system through life-cycle

Re-factoring Process



Results

- AO-MDS used to develop and deploy 3 large scale NASA AO solutions in 3 years (KSC, SSC)
- Re-use delivered 5 time speed-up in modeling (involving over 10,000 domain elements)
- Over 80% of functionality remains generic
- Quest for Class B Safety Critical Certification of AO-MDS currently in progress
- APL validated using physics based simulation
- Demonstration included unanticipated failures
- AO-MDS solution performed at 100%
  - No false positives
  - No false negatives

Autonomous Space Settlement

National Aeronautics and Space Administration