



Techniques and Lessons from Fault Protection

NASA Workshop on Autonomy Validation

John C. Day

Group Supervisor, Autonomy and Fault Protection (313I)

Jet Propulsion Laboratory, California Institute of Technology

818.354.2026

John.C.Day@jpl.nasa.gov



- **Background – Description of Fault Protection**
- **Fault Protection Design Challenges**
- **Perspective #1 – Coverage of Failure Space**
- **Perspective #2 – Application of Diagnosis Concepts**

Background: What is Fault Protection?

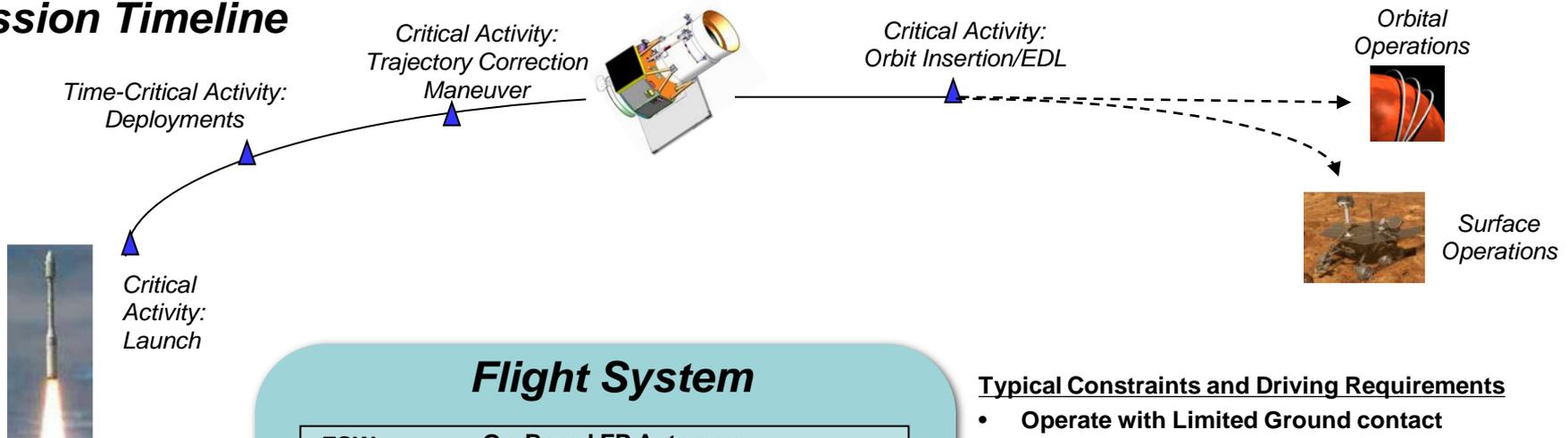


- **As used and applied at JPL, Fault Protection is both:**
 - A specific SE discipline (similar to EEIS or mission planning), whose activities are separately scheduled and tracked, and
 - The elements of a system that address off-nominal behavior
- **Focused on the flight system, Fault Protection includes**
 - Flight system fault detection and response
 - Ground-based failure diagnosis and recovery
 - Ground-based contingency planning and action

Background: Fault Protection Context



Mission Timeline

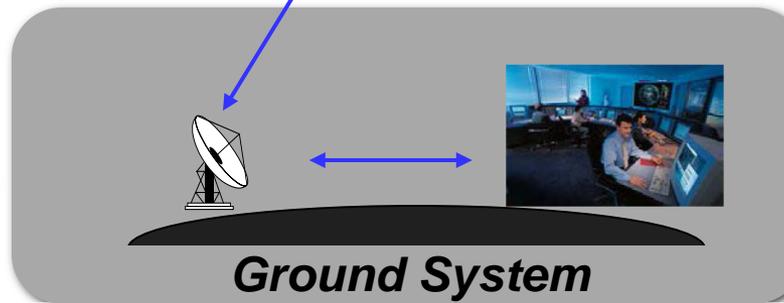
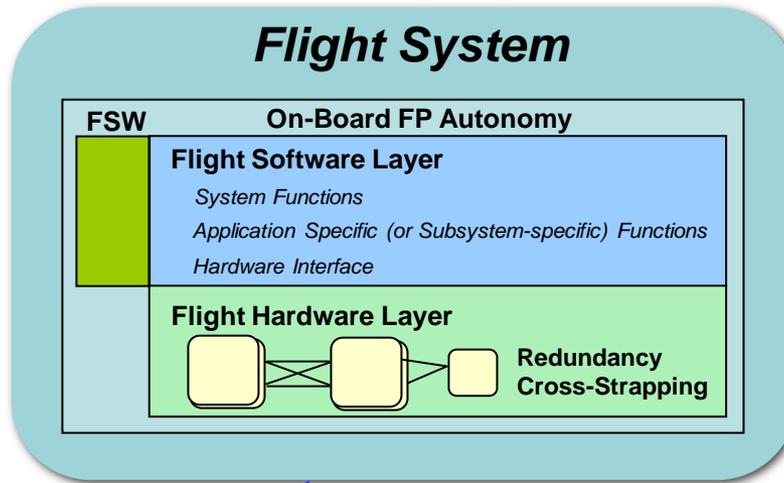


Flight System FDIR

- FSW Layers
 - Detect/Respond
- Hardware Layer
 - Detect/Respond

Ground FDIR

- * Monitor/Trend
- * Diagnosis/Recovery
- * Contingency Procedures
- * Test-bed/Simulation



Typical Constraints and Driving Requirements

- **Operate with Limited Ground contact**
 - Extended periods with no planned contact (1 to 4 weeks)
 - Planned contact periods may be short (1 to 2 hours)
 - Ground-based facilities may not support planned contacts (5% to 10%)
 - Large one-way light times (minutes to hours)
 - Low downlink data rates (10 to 40 bps)
- **Protect fragile elements of systems**
- **Leverage existing flight system components**
- **Protect/complete critical activities**
 - Orbit insertion, entry/descent/landing, irreversible deployments
- **Long mission life**
 - Survive *without maintenance* for primary missions lasting 5-11 years
- **Harsh environments**
 - Total Ionizing Dose of 100 krad to 4 mrad



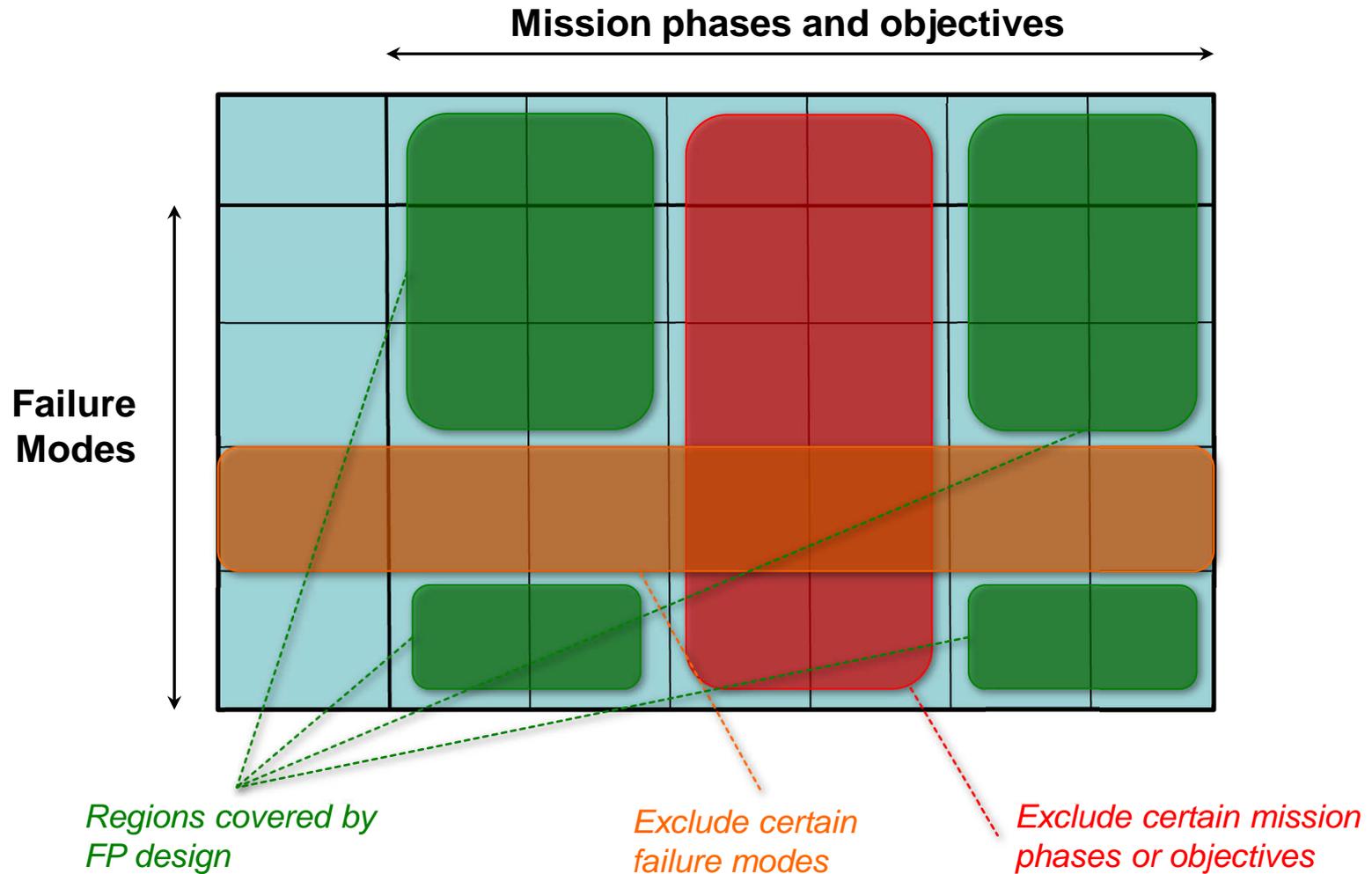
- **Challenges**

- Failure space is essentially infinite
- Environment can differ from predictions
- Limited physical redundancy (due to constraints on mass/power/etc.)

- **Solutions**

- Focus on “function preservation”, instead of “fault protection”
 - Finite, tractable – as opposed to infinite, unknown
- Establish “safety nets” for core functions
 - Account for unknown failure causes in key areas
- Design margin and flexibility into system (functional redundancy, ability to add/modify functions and allocations)
- Limit set of failure causes assessed/covered; typical exclusions are:
 - Common-mode failures
 - Unexposed design flaws
 - Operator error

Illustration of Limiting Failure Space



Perspective #1 – Coverage of Failure Space

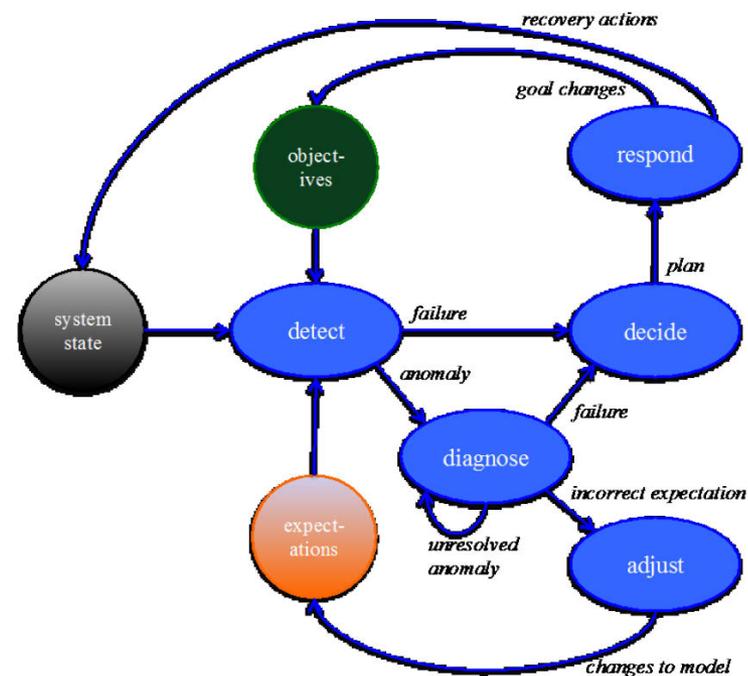


- **Can the coverage of “failure space” by a FP design provide insights into how to provide test coverage by a validation suite?**
- **FP Coverage**
 - Identify “failure space” (from objectives and failure modes)
 - Define regions of failure space that will not be covered
 - Common-cause failures, design failures
 - Identify core functions, and ensure they are protected, regardless of cause (safety nets)
 - Omit particular failure cases that are low risk or exceptionally unlikely
- **Validation Suite Coverage**
 - Identify scenario space
 - Define regions of scenario space that will not be covered by validation
 - Identify properties of interest, and develop tests or analyses that ensure they are validated, regardless of the scenario/case
 - Use performance of core function as criteria?
 - Omit particular scenarios that are low risk or exceptionally unlikely

Perspective #2 – Application of Diagnosis



- **Diagnosis is a class of functions that, based on detected anomalies, determines whether the anomaly is evidence of:**
 - A previously-unobserved failure, or
 - An incorrect expectation of behavior
- **A successful diagnosis also provides the location of and reason for the detected anomaly**
- **Changes to the system or system model can then be made to react to the anomaly**
- **Can this view of diagnosis be applied to constructing a validation suite?**





- **If validation can be accurately construed as “does a system perform as expected?”, then diagnosis can be used as real-time validation of system performance.**
 - Answers the “why” question when an anomaly occurs
- **In system validation, need to determine whether the anomaly is evidence of:**
 - A previously-unobserved failure, or
 - An incorrect expectation of behavior
- **A successful diagnosis also provides the location of and reason for the detected anomaly**
- **For identified instances of unexpected performance:**
 - If there is a flaw in the system, it needs to be repaired, mitigated, or used as-is
 - If there is an error in the expectations, then either changes to the current scenario are needed, and/or new scenarios included
 - Use protection of core function as a criterion for changes



BACKUP

Covering the “Failure Space”



Top-down
assessment

FP necessary to maintain acceptable functionality for each identified failure scenario

determine system functions

functional analysis, FTA, HA, IHA

determine states associated with each function

identify state(s) associated with each function

determine acceptable ranges

determine the acceptable values of each state for relevant mission phases/activities (goals); acceptable values may change over course of mission

analyze set of success scenarios

for each mission phase/activity, determine FDIR necessary to maintain acceptable function

Develop necessary FP

analyze set of failure scenarios

for each failure scenario, assess effectiveness

determine set of failure scenarios

for each failure effect, assess relevant mission phases/activities; add identified hazards

determine set of failure effects

for each failure mode, identify failure effects

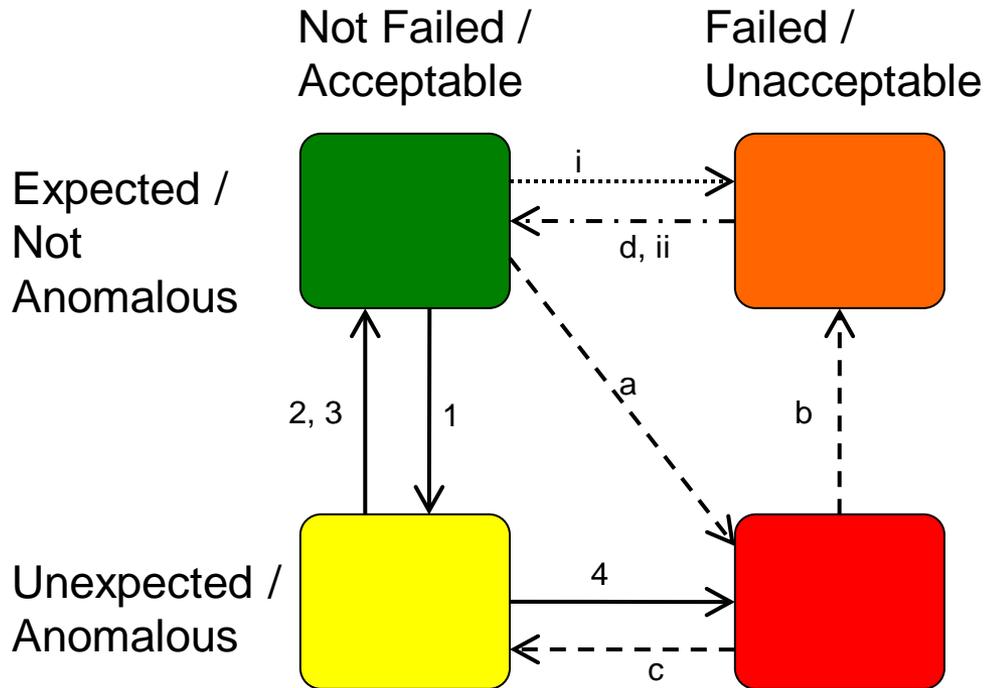
determine fault set

FMEA, FTA

FP necessary to maintain acceptable functionality through all mission phases

Bottom-up
assessment

Progression of Anomalous/Failed States



Anomaly, no Failure

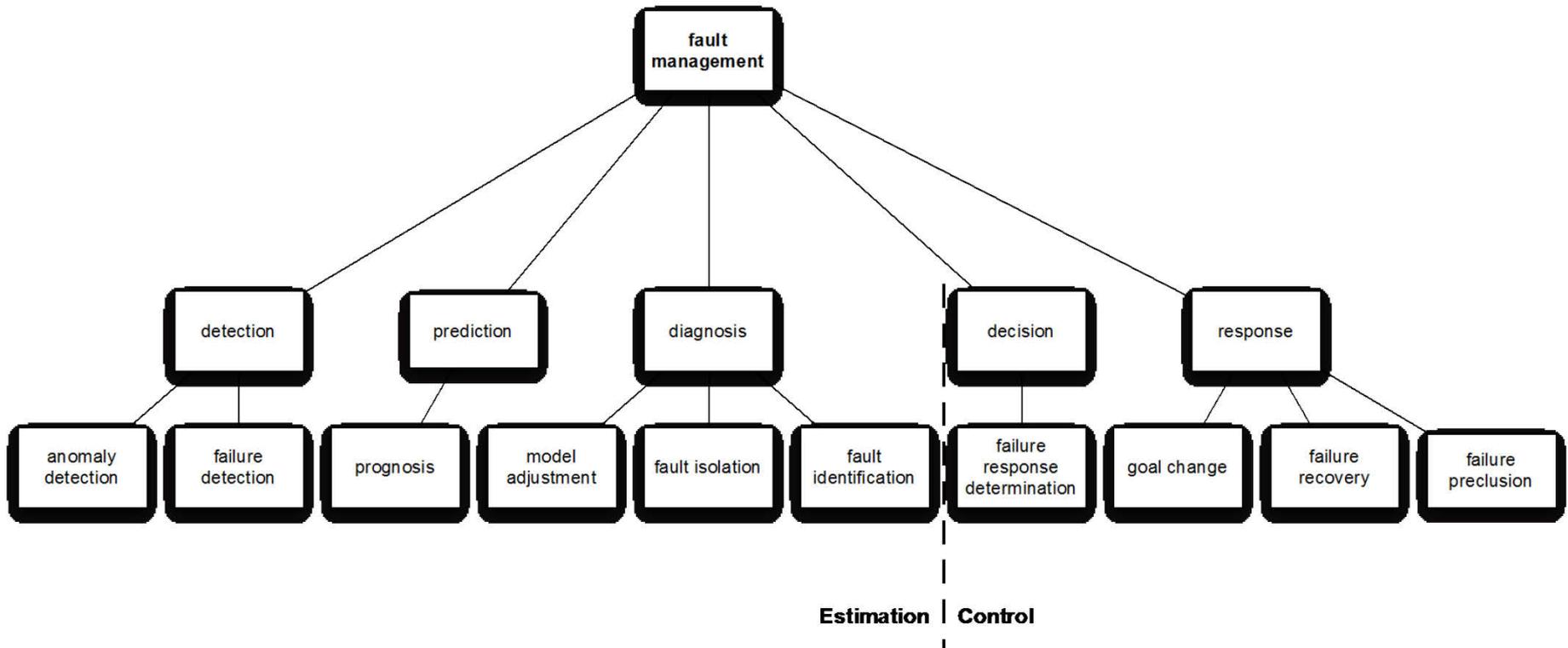
- 1) current value of state reaches an unexpected value
 - 2) review of system data indicates that model/expectation is invalid, and state is expected (expectations changed) [e.g., noise in RF link due to un-modeled effect]
- model reviewed and parameters adjusted until model predicts current behavior (e.g., if RWA unhealthy, will have larger attitude errors)
 - review of system data indicates that this is an unacceptable value (indicative of a failure; the goal is adjusted)

Anomaly, with Failure

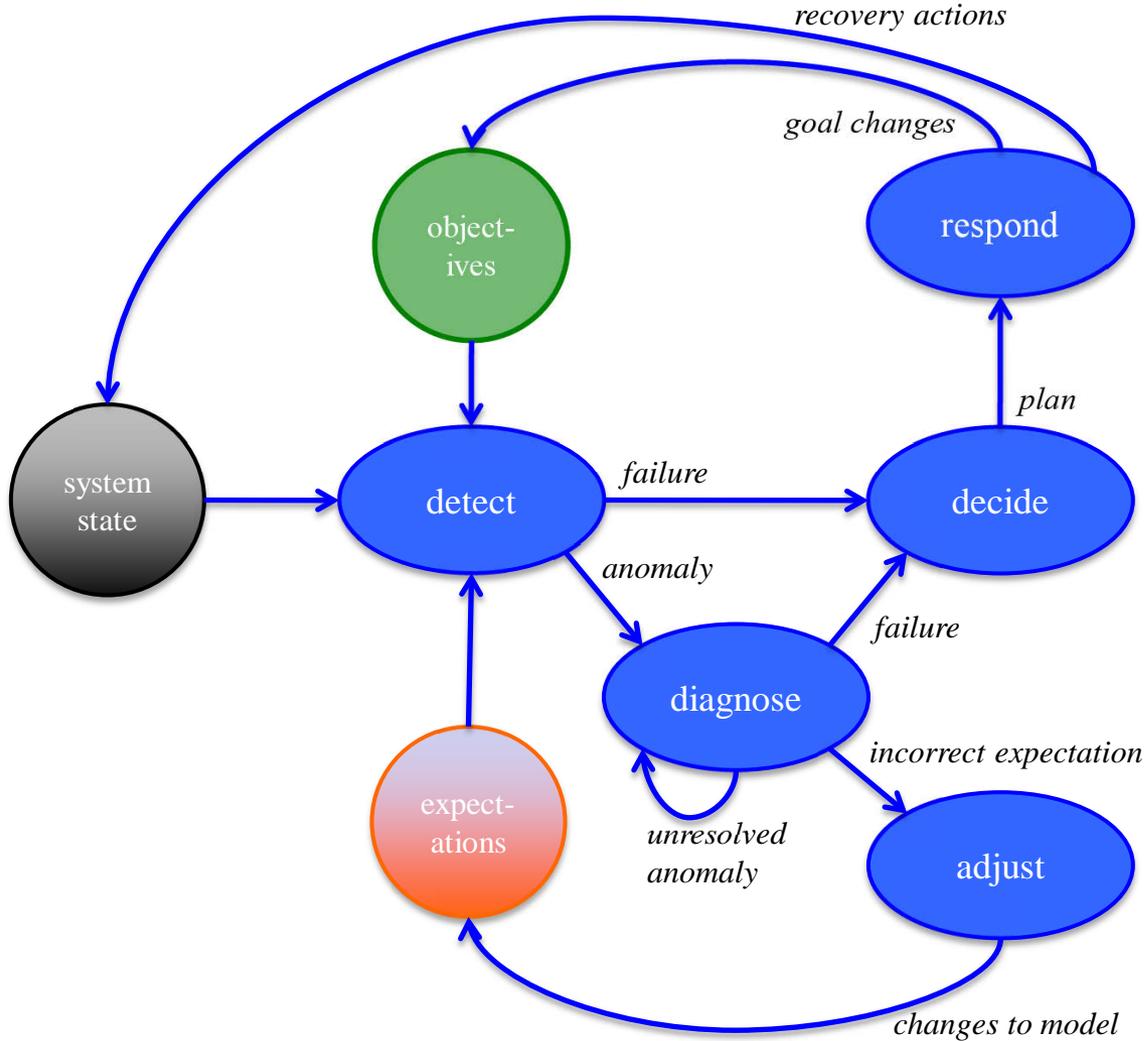
- a) current value of state unexpectedly reaches an unacceptable value
 - b) model reviewed and parameters adjusted until model predicts current behavior (e.g., if IMU1 unhealthy, will have attitude failure)
- review of system data indicates that model/expectation is invalid, and state is acceptable (expectations changed)
 - recover intended functionality by restoring state to acceptable value and/or changing functional goal

Failure, no Anomaly

- i. expected condition results in failure
- ii. recover intended functionality by restoring state to acceptable value and/or changing functional goal



Simplified Fault Management Loop



System States – Failure Modes and Objectives

