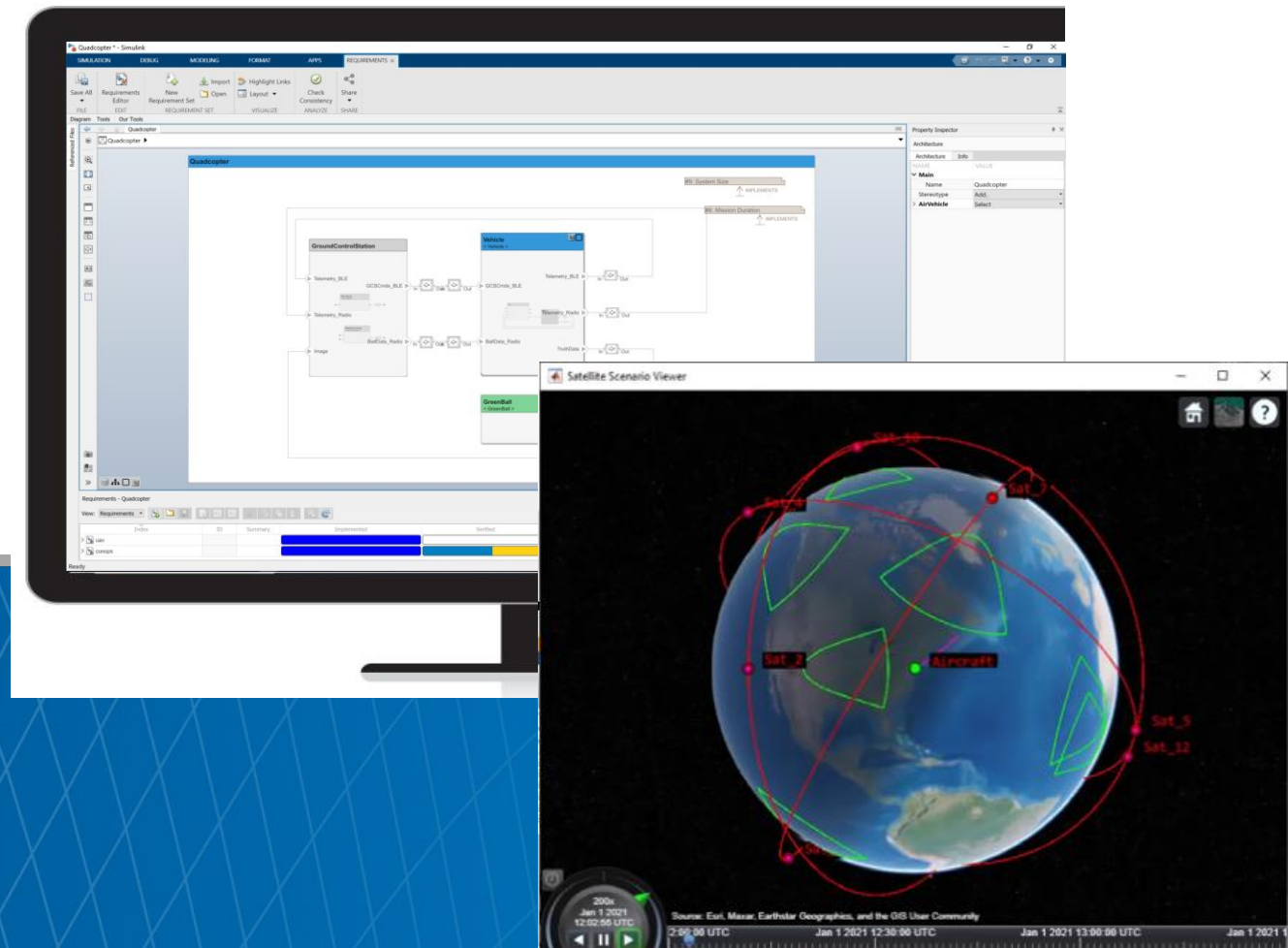


Model Based Systems Engineering for Space Based Applications



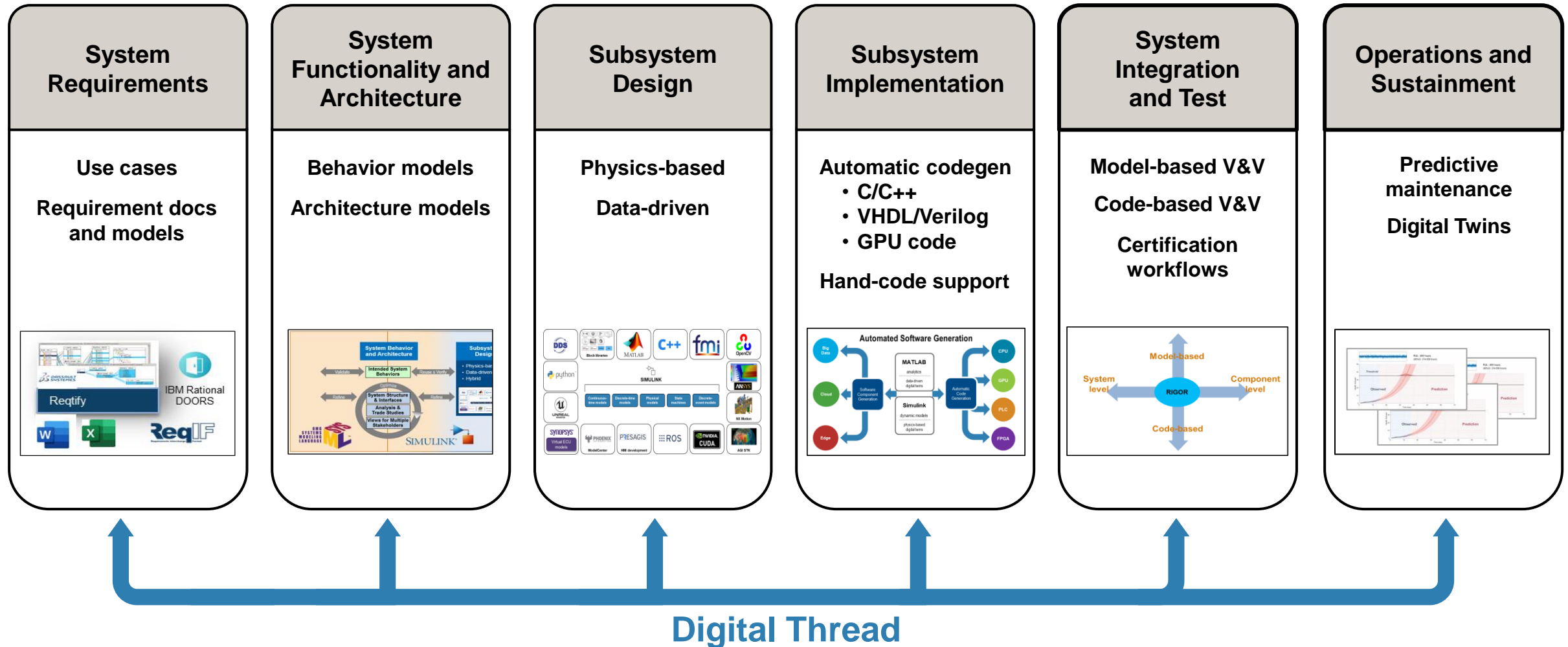
Andrew Grabowski
Application Engineer
MathWorks – Washington, D.C



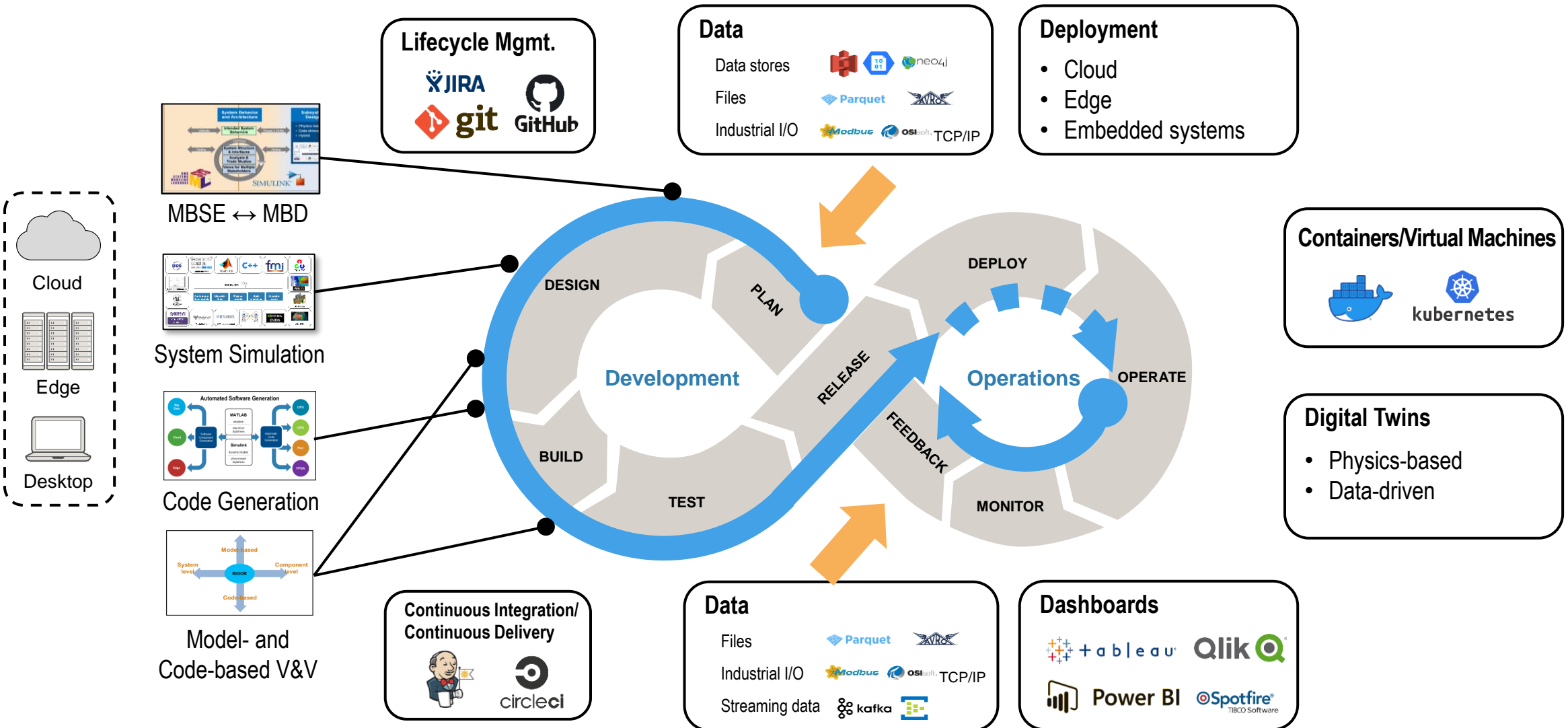
Key Takeaways

- MATLAB and Simulink products for Model Based System Engineering (MBSE)...
 - enable intuitive, scalable and adaptive modeling of architecture models
 - utilize the architectures to conduct analysis early in the design cycle
 - provide a digital thread from architecture to behavior models

MathWorks Digital Engineering Capabilities

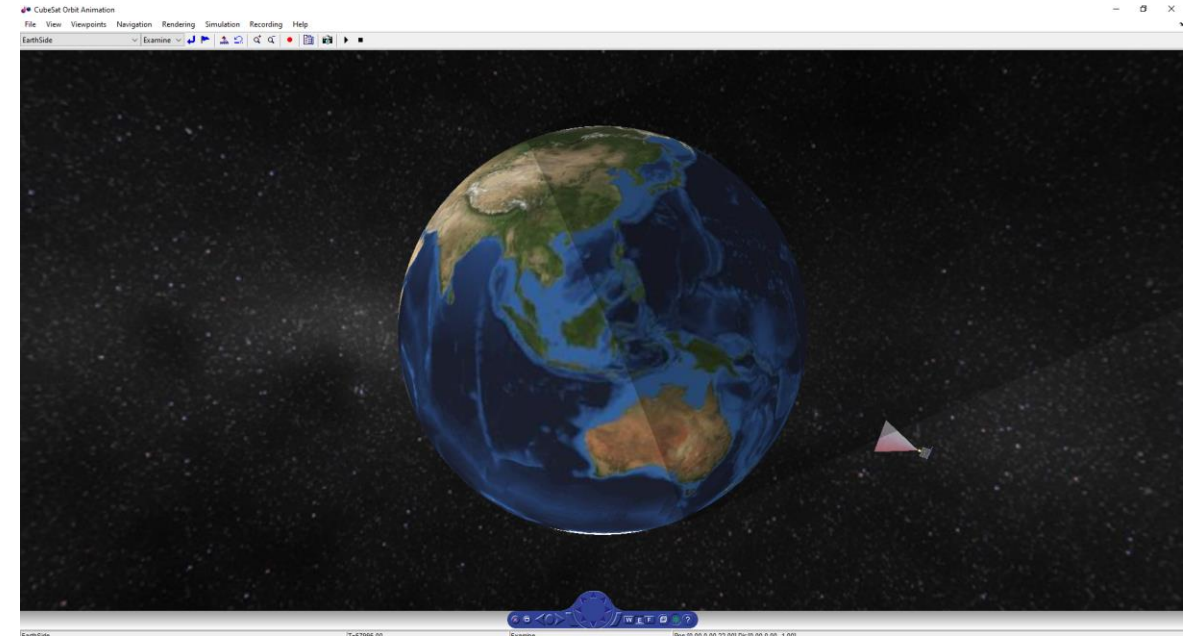


Software and Systems Integrated Workflow



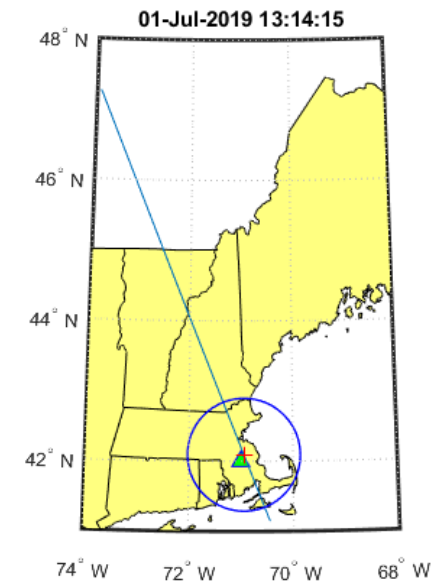
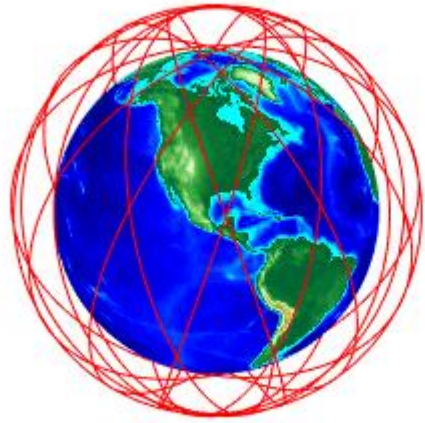
Agenda

- **Explore CubeSat Mission System**
- **Discuss and Demo Model Based System Engineering**
- **Choose your own adventure 😊**



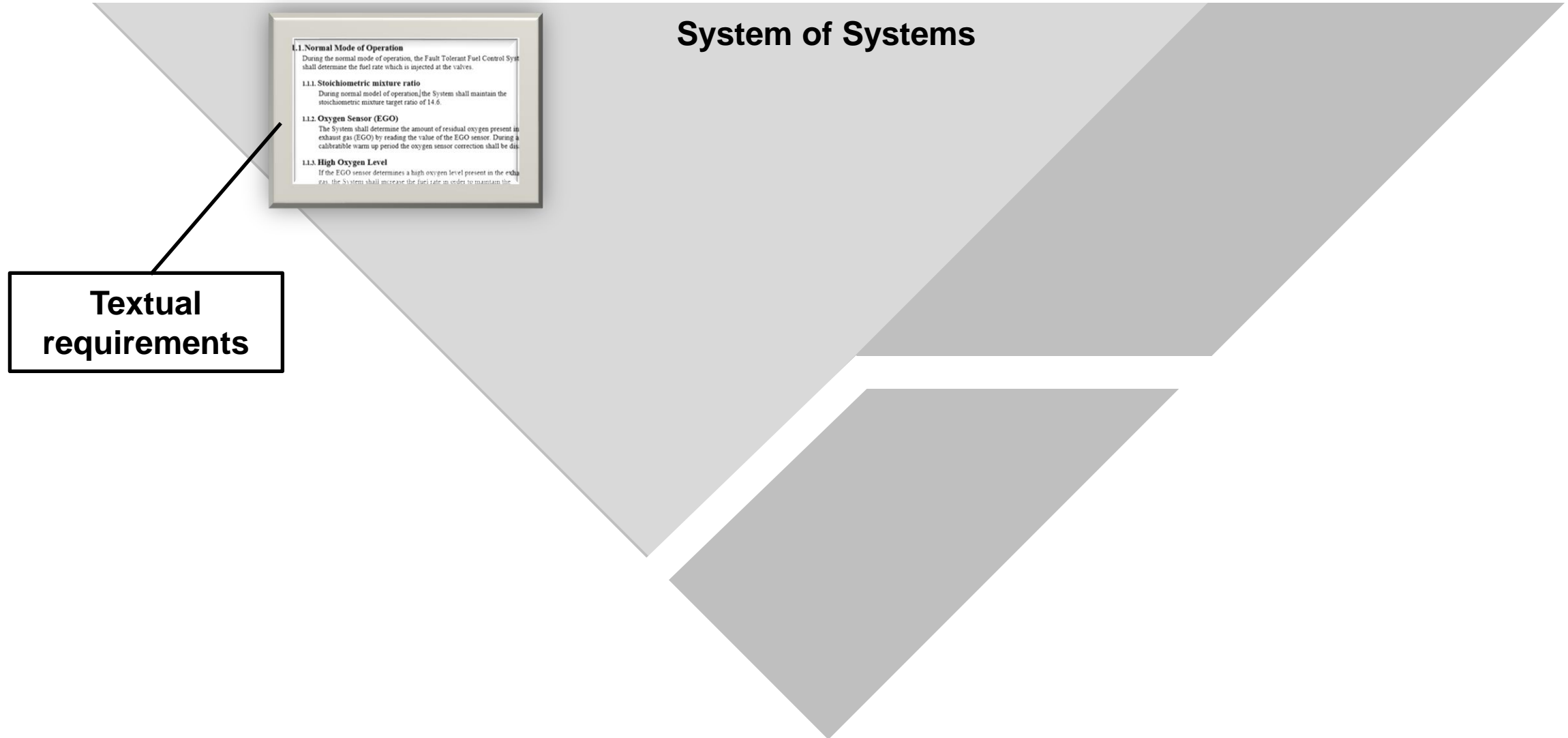
High Level Requirements/ Concept of Operations

- The system shall provide and store visual imagery of MathWorks headquarters [42.2775 N, 71.2468 W] 1 time daily at 10 meters resolution.
 - The satellite shall provide for collection by the ground system visual imagery of MathWorks headquarters 1 time daily at 10 meters resolution.



System Composer OMG Satellite Model and Mission Analysis

How is this done today?



System

1.1. Normal Mode of Operation

During the normal mode of operation, the Fault Tolerant Fuel Control System shall determine the fuel rate which is injected at the valves.

1.1.1. Stoichiometric mixture ratio

During normal mode of operation, the System shall maintain the stoichiometric mixture target ratio of 14.6.

1.1.2. Oxygen Sensor (EGO)

The System shall determine the amount of residual oxygen present in exhaust gas (EGO) by reading the value of the EGO sensor. During a calibratable warm up period the oxygen sensor correction shall be disabled.

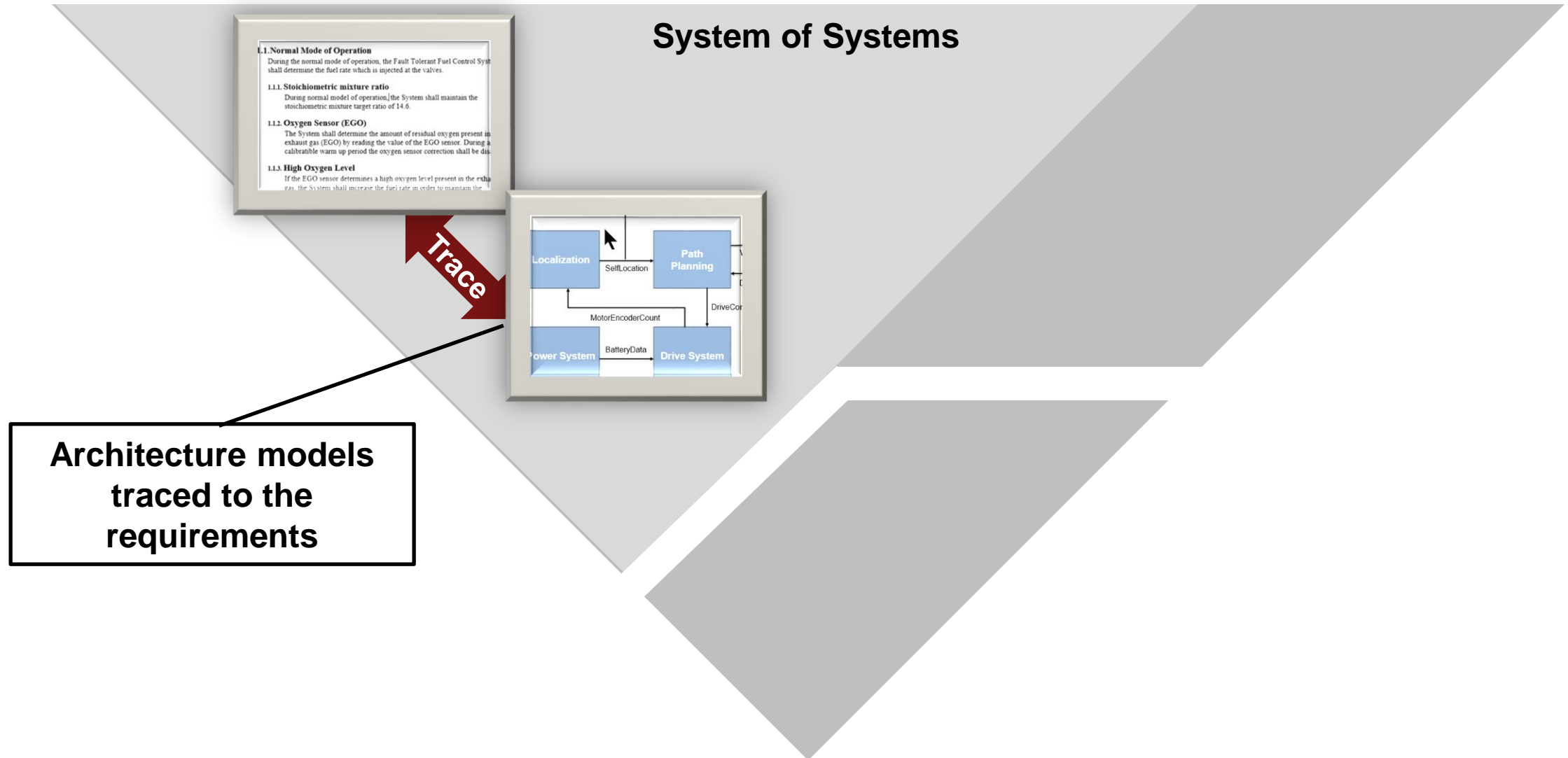
1.1.3. High Oxygen Level

If the EGO sensor determines a high oxygen level present in the exhaust gas, the System shall increase the fuel rate in order to maintain the

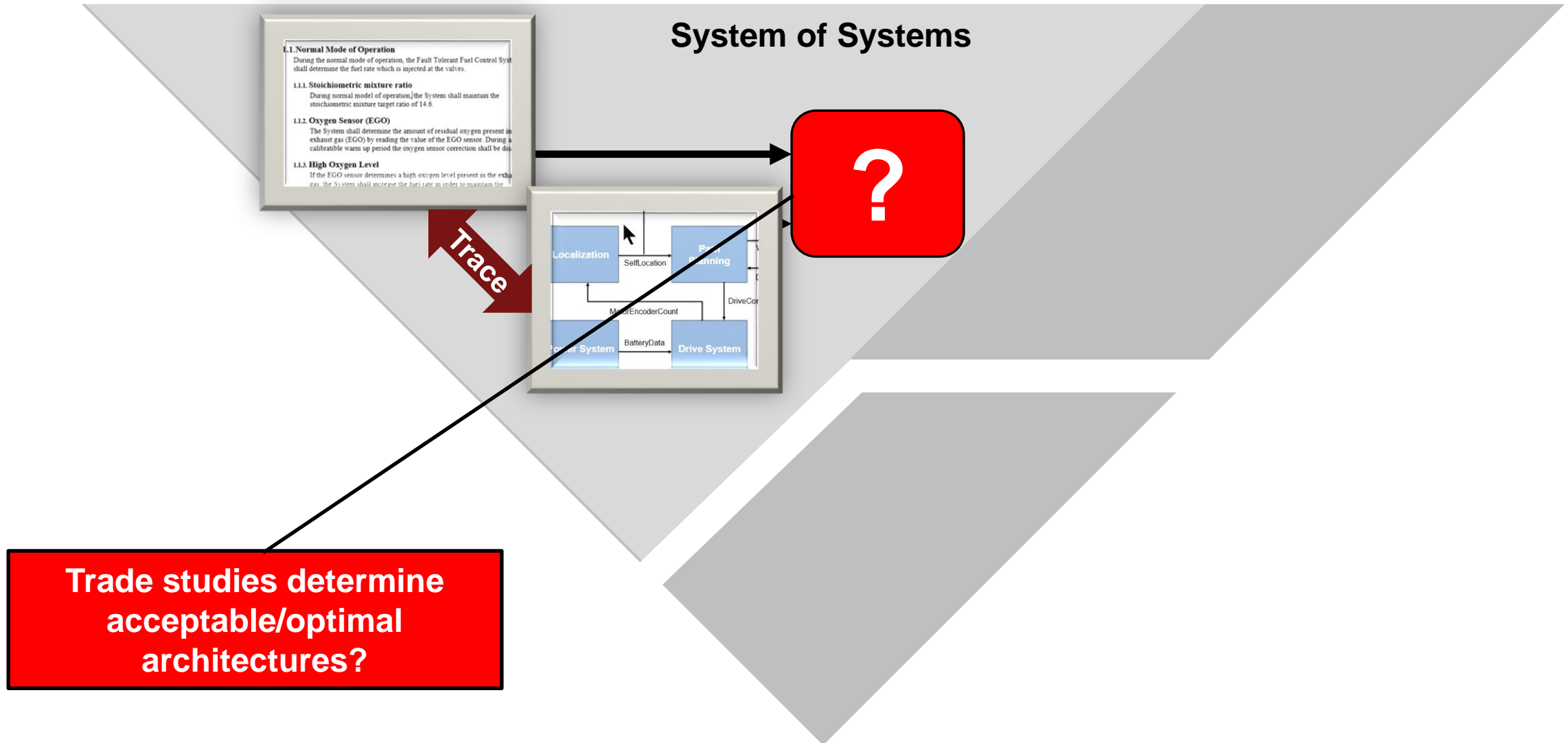
```
graph TD
    DRIVER[DRIVER] --> APP[APP]
    APP -- COMMAND --> SUPERVISOR[SUPERVISOR]
    SUPERVISOR --> APP
    SUPERVISOR --> SIGNAL_PROCESSING[SIGNAL PROCESSING]
    SIGNAL_PROCESSING --> TRACKING[TRACKING]
    TRACKING --> TARGET_VAL[TARGET VAL]
    TARGET_VAL --> SUPERVISOR
    TRACKING --> VALVE_CONTROL[VALVE CONTROL]
    VALVE_CONTROL --> VALVES[VALVES]
```



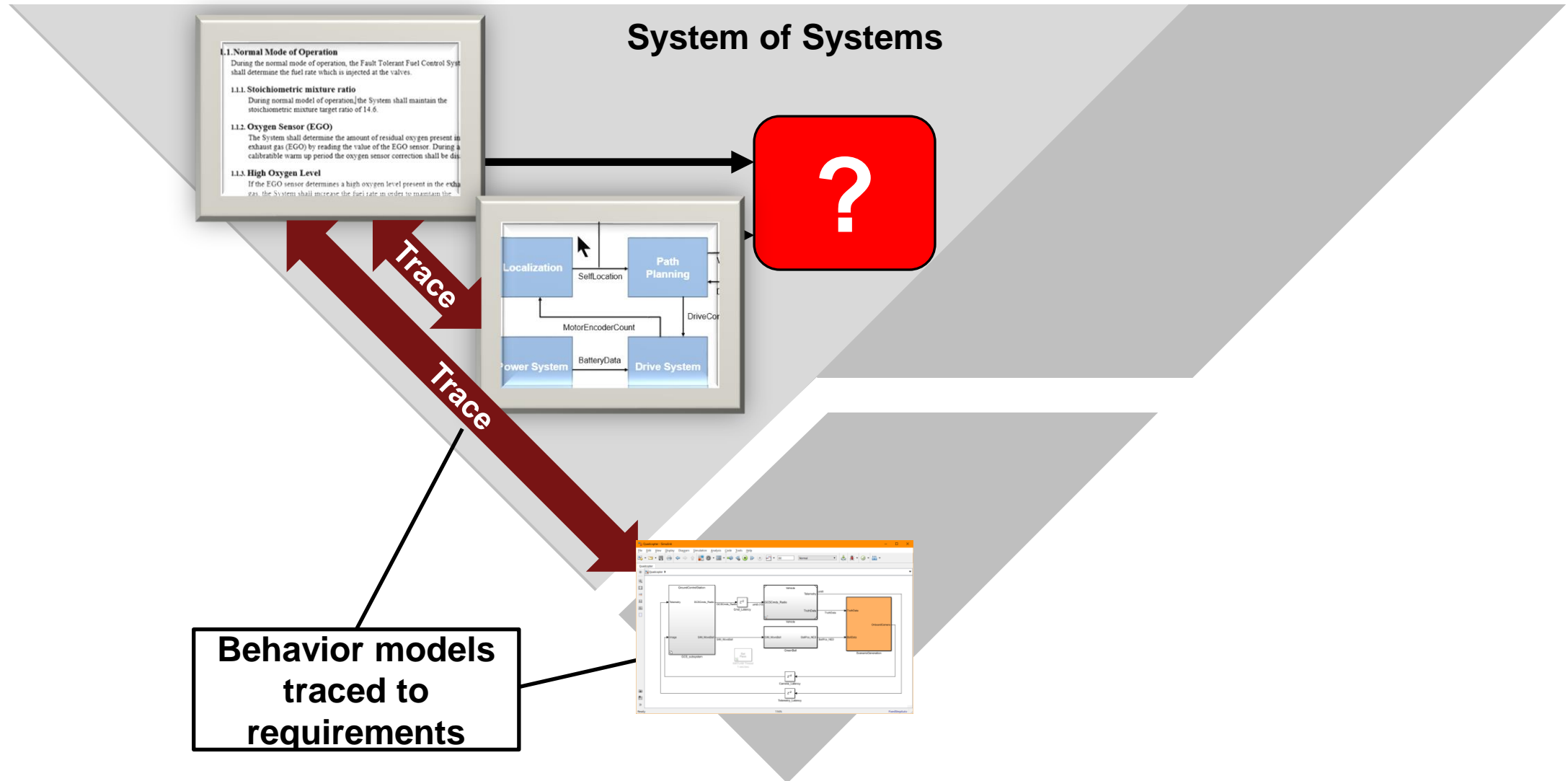
How is this done today?



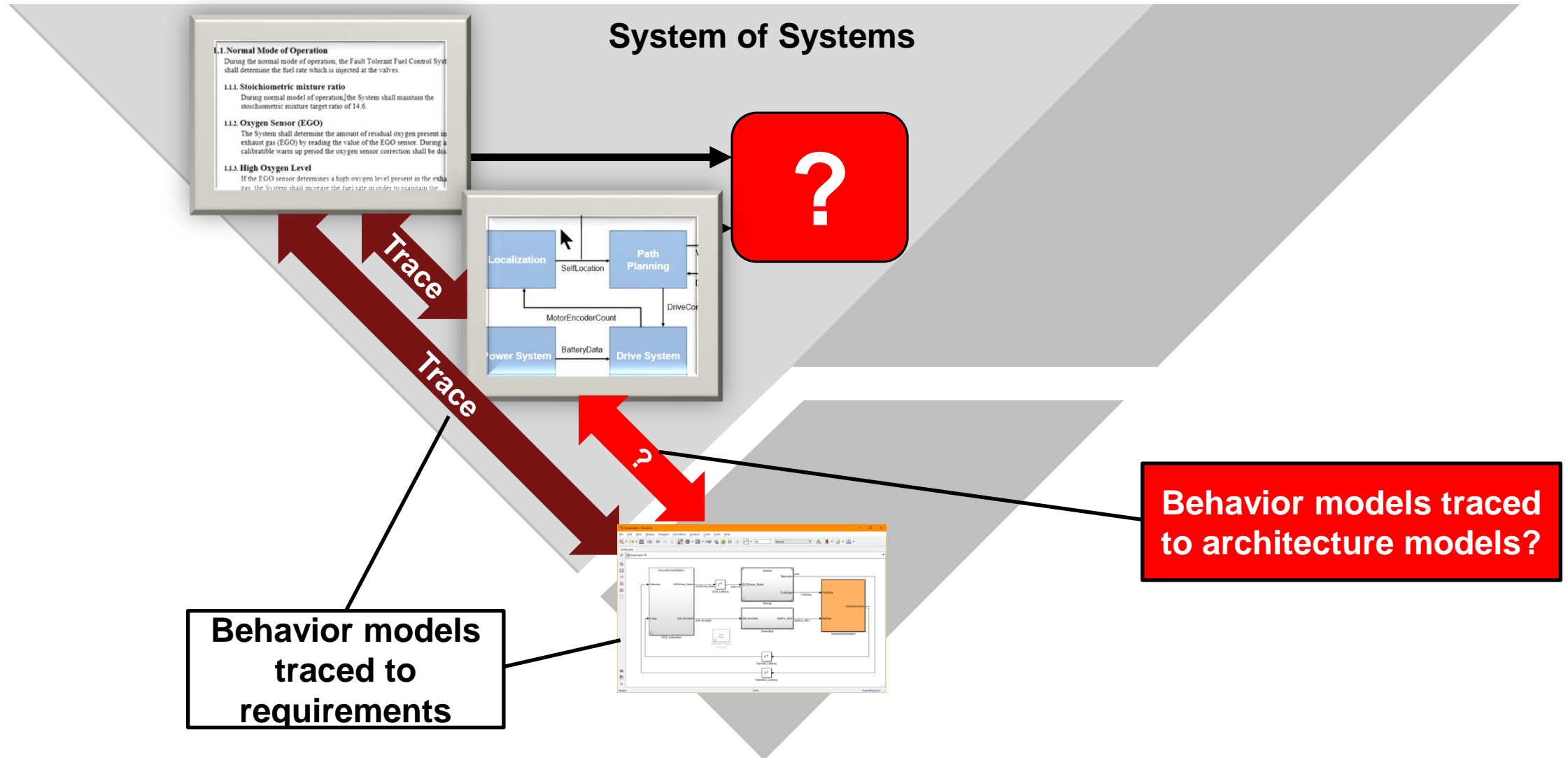
How is this done today?



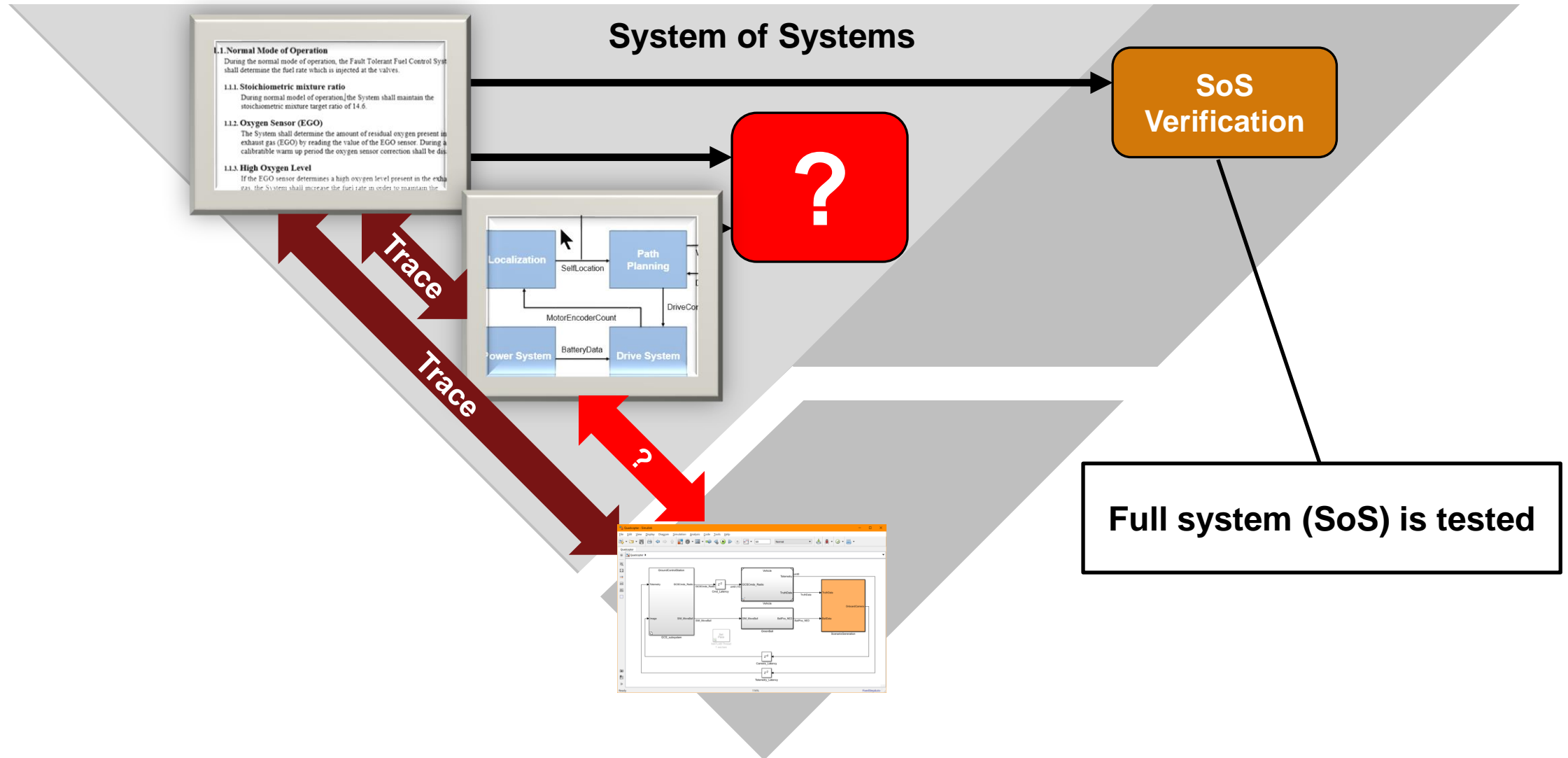
How is this done today?



How is this done today?



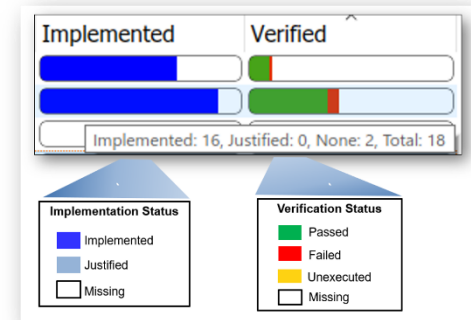
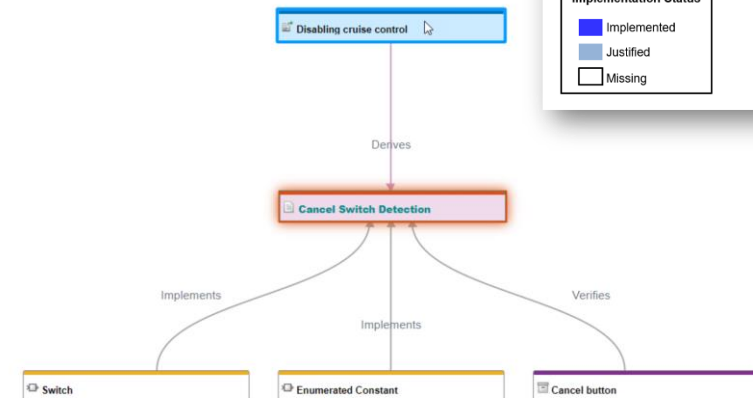
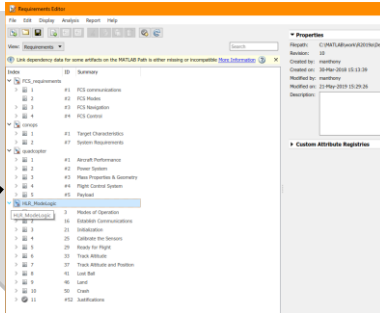
How is this done today?



How does MathWorks do it?

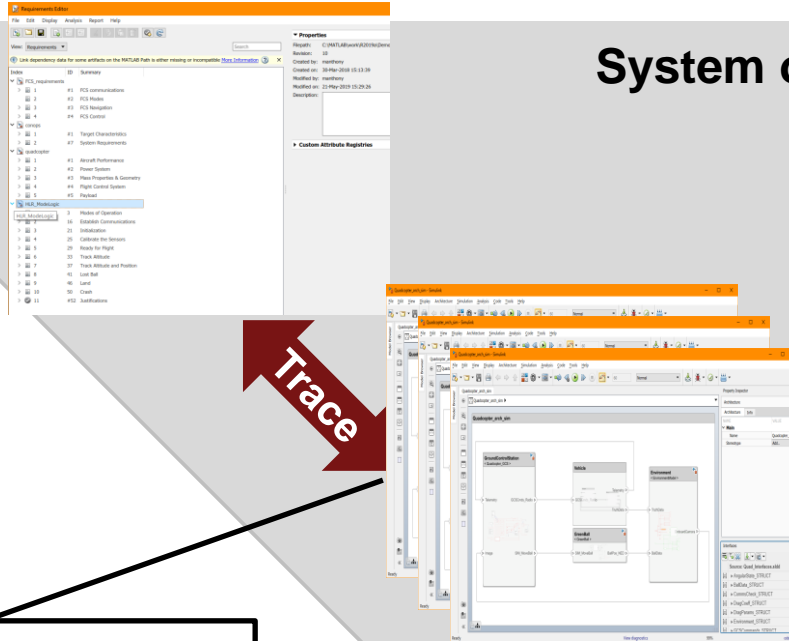
System of Systems

Textual requirements

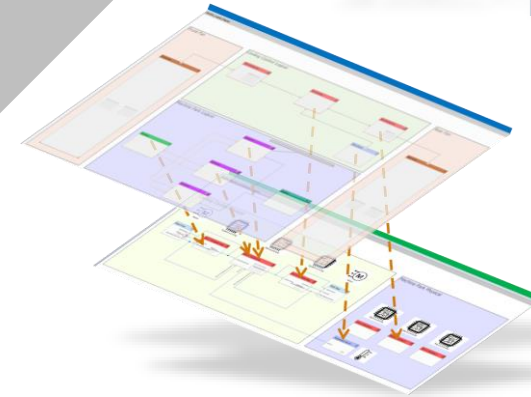
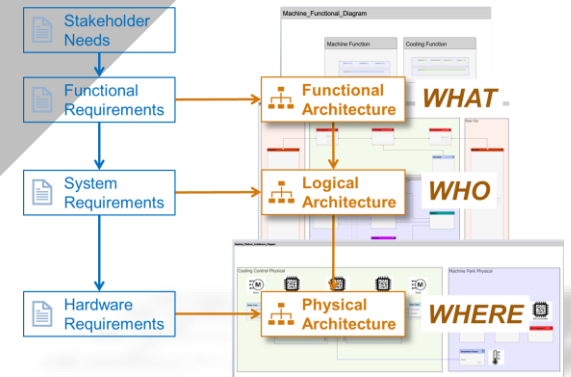


How can MathWorks help?

System of Systems



Architecture models
traced to the
requirements



How is analysis done today?



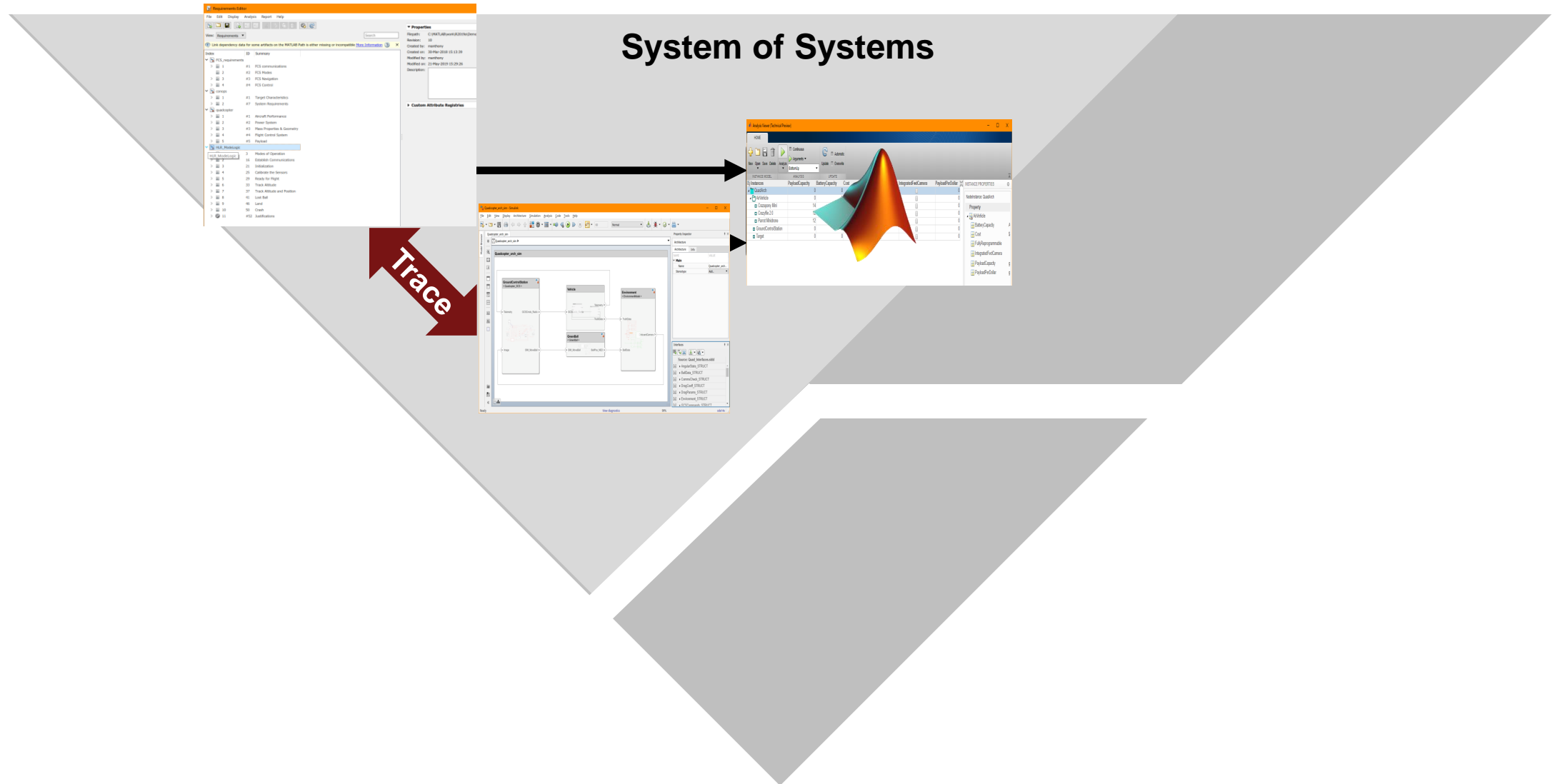
Typical Tools:

- MATLAB
- Excel
- In house / 3rd Party analysis tools

Pain Points:

- Disjoint analysis tools
 - Difficult to bring in quantitative Systems engineering methodology (Pugh, Analytic Hierarchy, Kepler Tregoe)
- Multiple tools needed to collect data and report out
- Verbal knowledge transfer
- System Level Verification

How can MathWorks help?



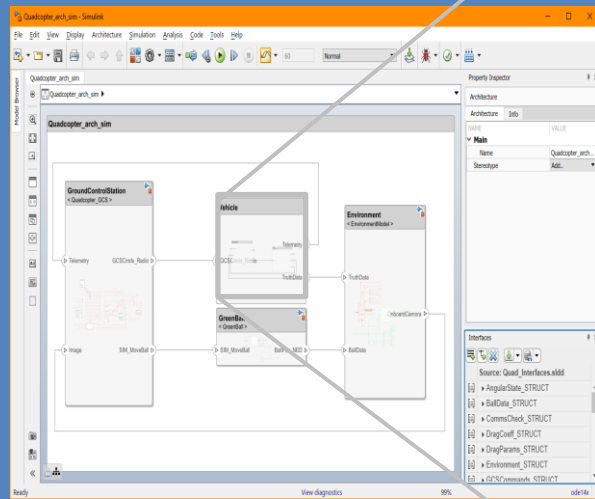
The diagram illustrates the internal structure of the **PowerSystemPlant** component. It consists of two sub-components: **SolarPanel** and **Battery**.

- SolarPanel** has two provided interfaces: `attitudeStates` and `actuatorCmd`. It also has a required interface `powerIn` (indicated by a hollow triangle).
- Battery** has two provided interfaces: `powerStates` and `sensorData`. It also has a provided interface `powerIn` (indicated by a solid line with a hollow triangle).

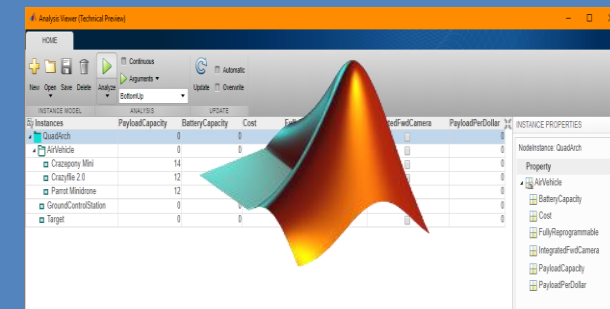
A dependency arrow points from the `powerIn` required interface of **SolarPanel** to the `powerIn` provided interface of **Battery**, indicating that the SolarPanel component depends on the Battery component for its powerIn functionality.

System Composer Supports Trade Study Formulation and Analysis

Architecture Model in System Composer

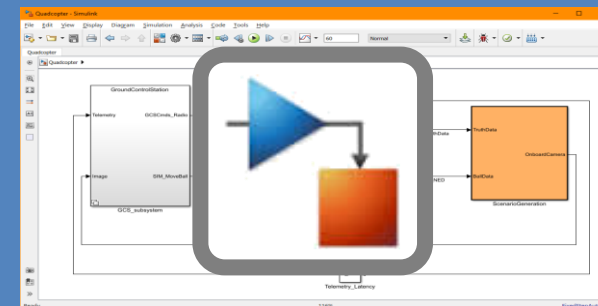


Trade Studies



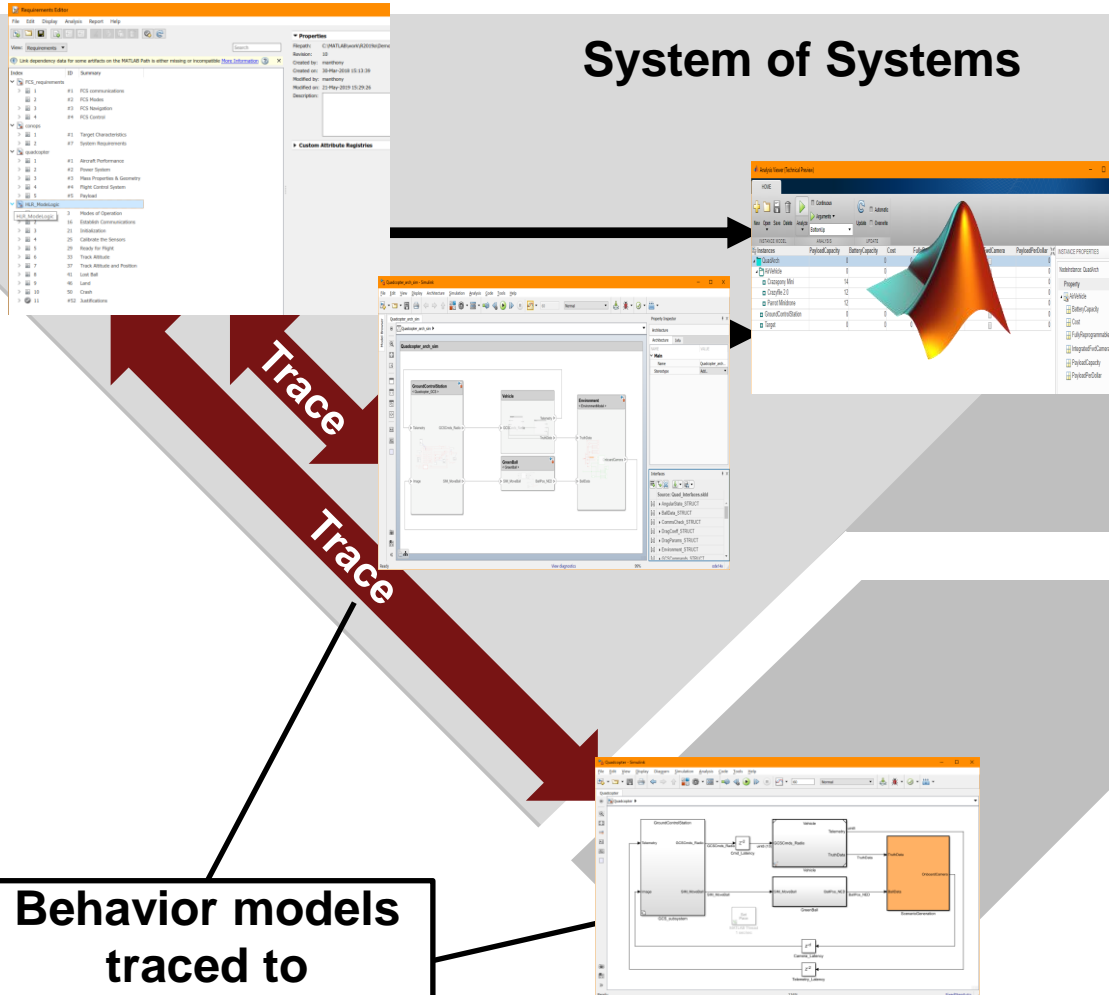
**Behavior Confirmation of Trade Studies
via Simulation**

Simulink Behavior Models



How can MathWorks help?

System of Systems



**Behavior models
traced to
requirements**

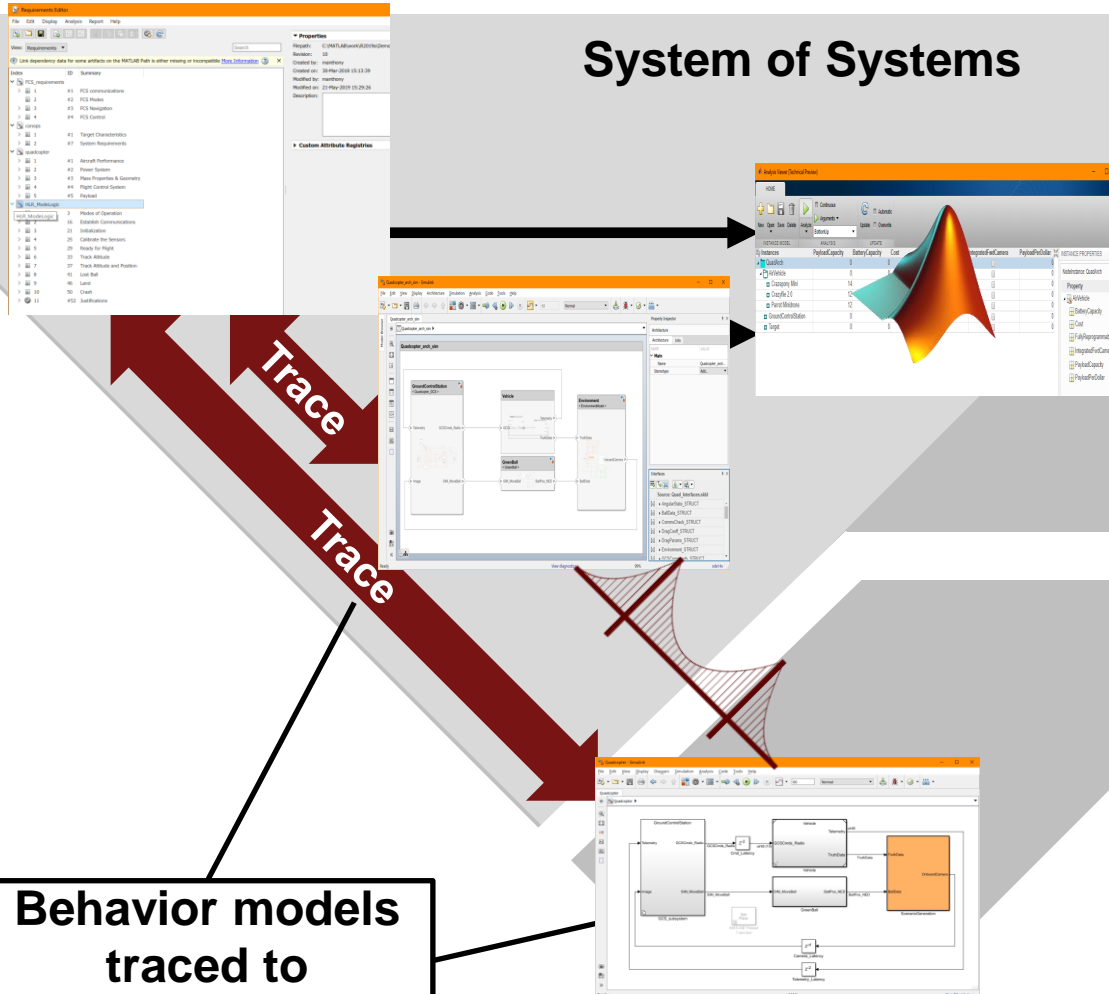
System of Systems

Behavior models traced to

Behavior models traced to requirements

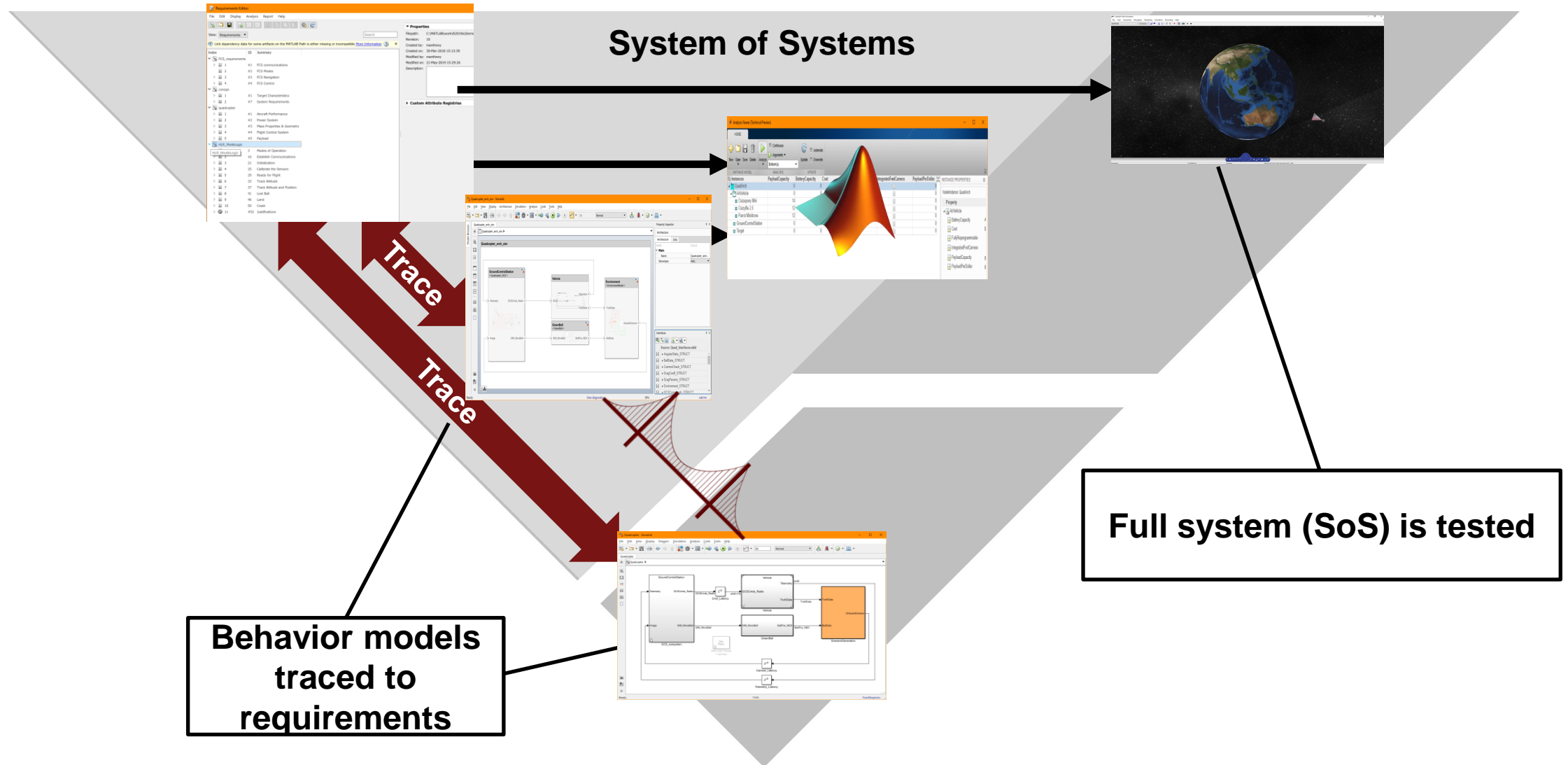
How can MathWorks help?

System of Systems

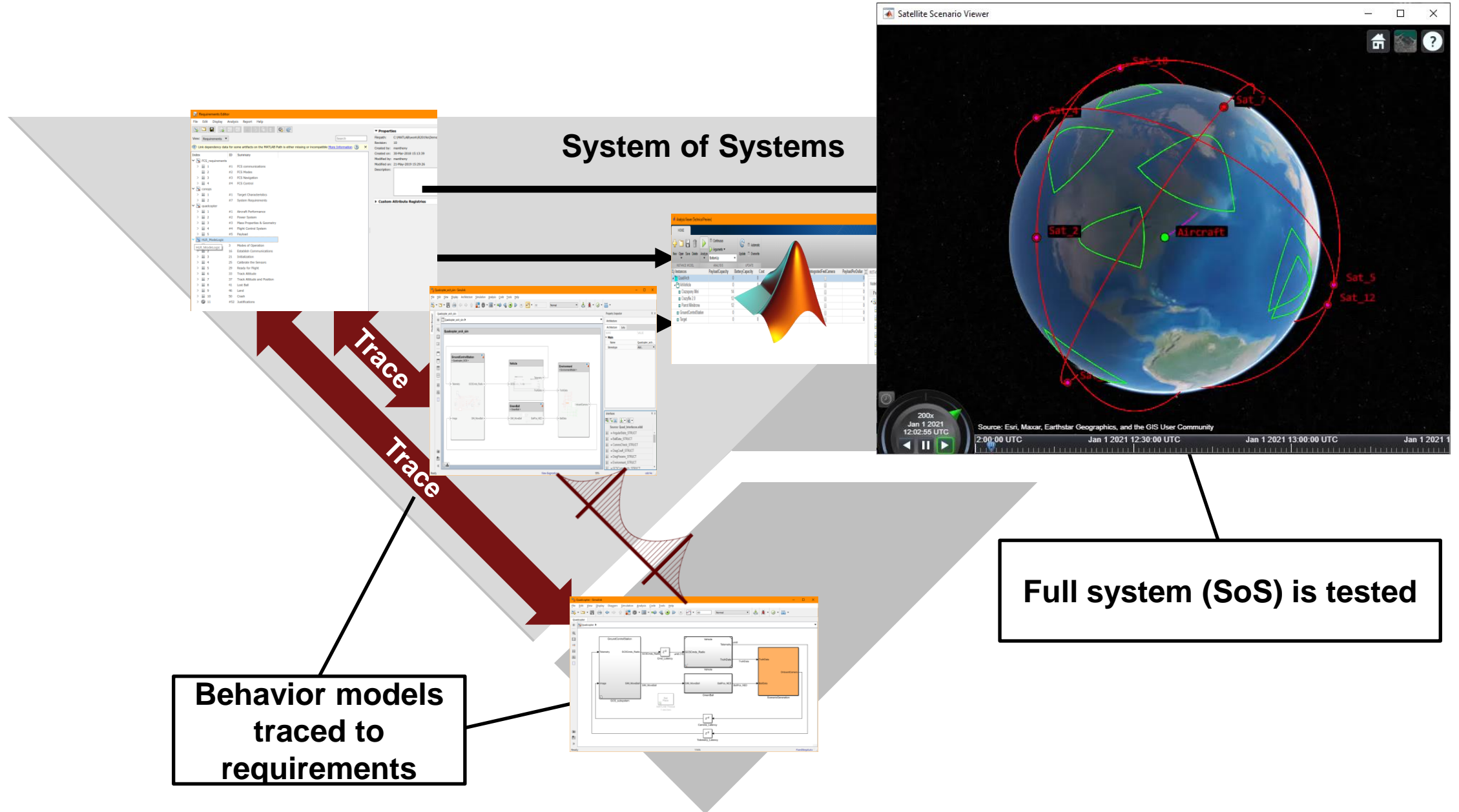


**Behavior models
traced to
requirements**

How can MathWorks help?



How can MathWorks help?

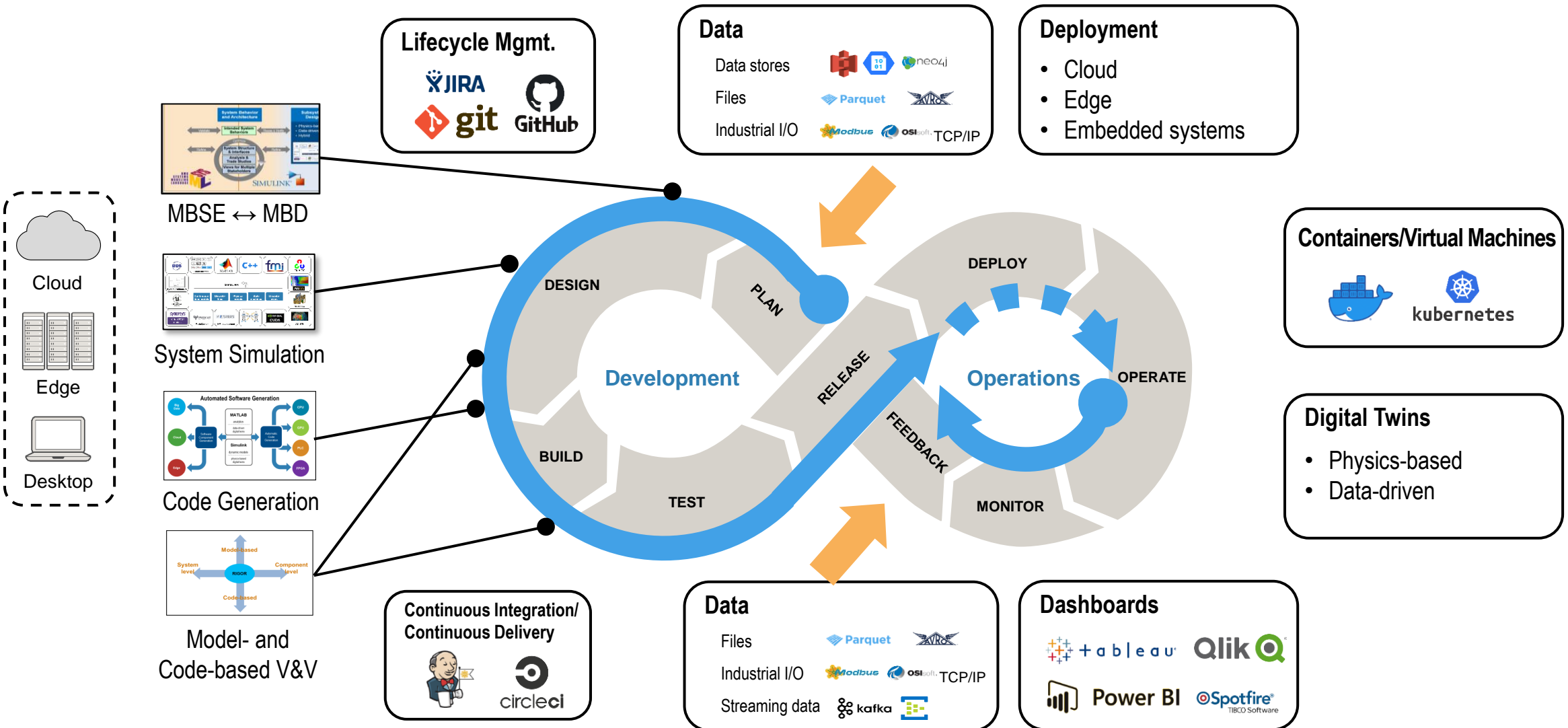


How can we start System Validation?

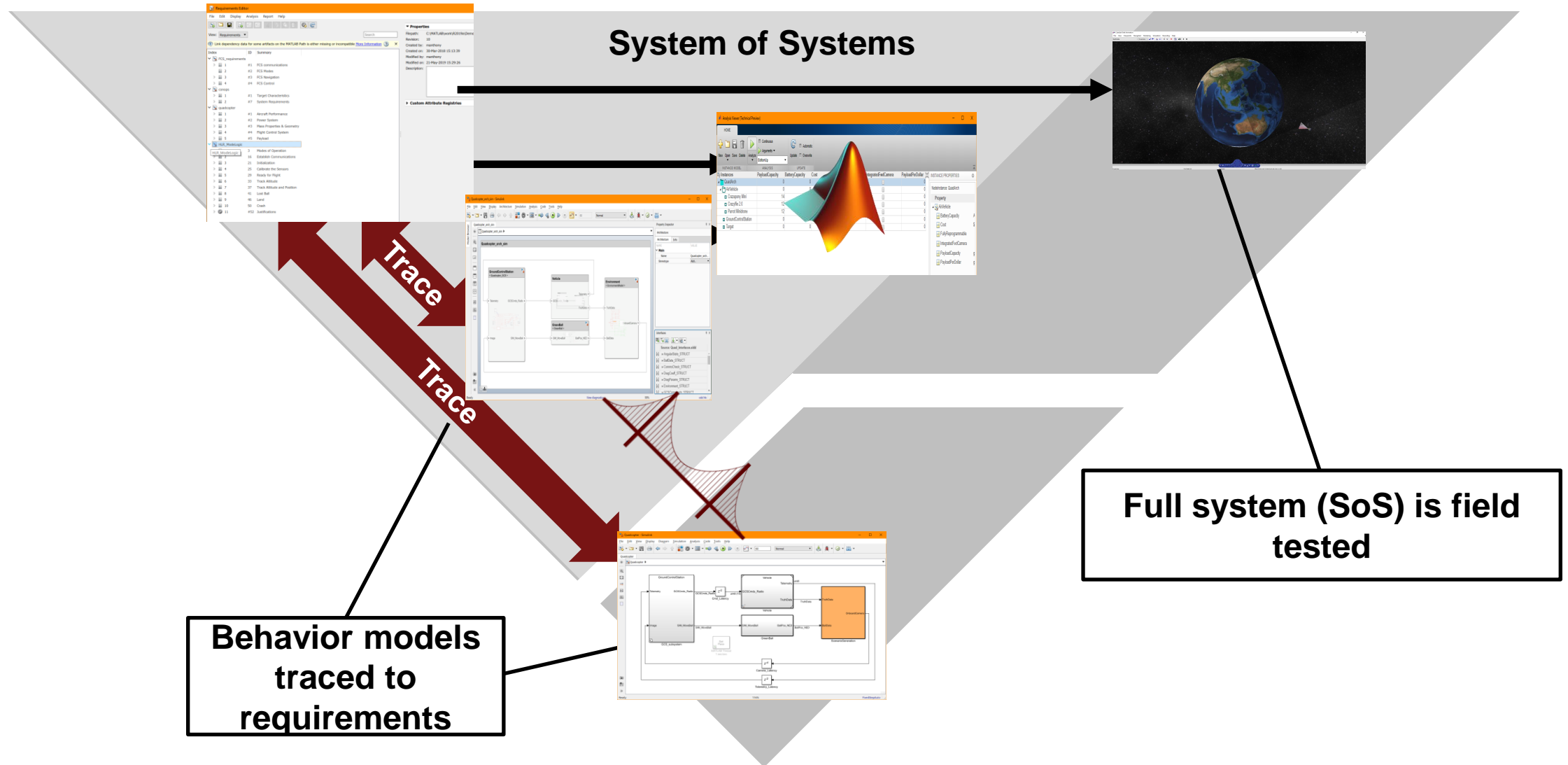
Key Takeaways

- MATLAB and Simulink products for Model Based System Engineering (MBSE)...
 - enable intuitive, scalable and adaptive modeling of architecture models
 - utilize the architectures to conduct trade studies early in the design cycle
 - provide a digital thread from architecture to behavior models

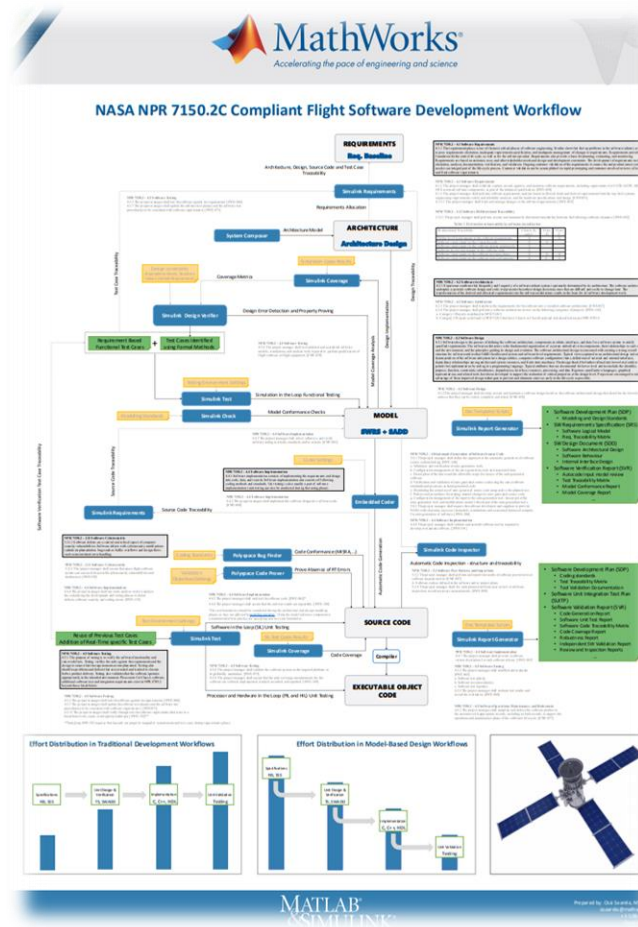
Software and Systems Integrated Workflow



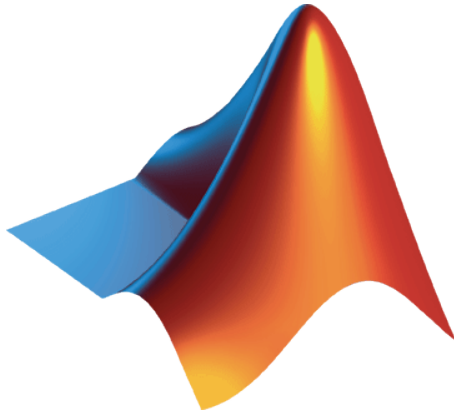
Summary



Interested in applying MathWorks tools to NPR 7150.2C...



<https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/aerospace-defense/standards/npr7150-2c-workflow.pdf>



Questions?

Andrew Grabowski
Application Engineer
agrabows@mathworks.com

MathWorks Support Mechanisms: Collaboration Ensures Success



Technical Support

- Product questions
- General support
- 508-647-7000



Application Engineering

- Product/Capability demonstrations
- Lunch-&-Learns, Workshops, WebEx's, etc.
- Evaluation support



Pilot Engineering

- Complimentary, guided support for adoption of new tools/processes
- Deep engagement evaluations/demonstrations



Training

- Paid training on specific tools and/or processes
- On-site, web-based instructor lead, & self-paced online



Consulting

- Paid engagements (custom targets, tool customization, advisory services)

