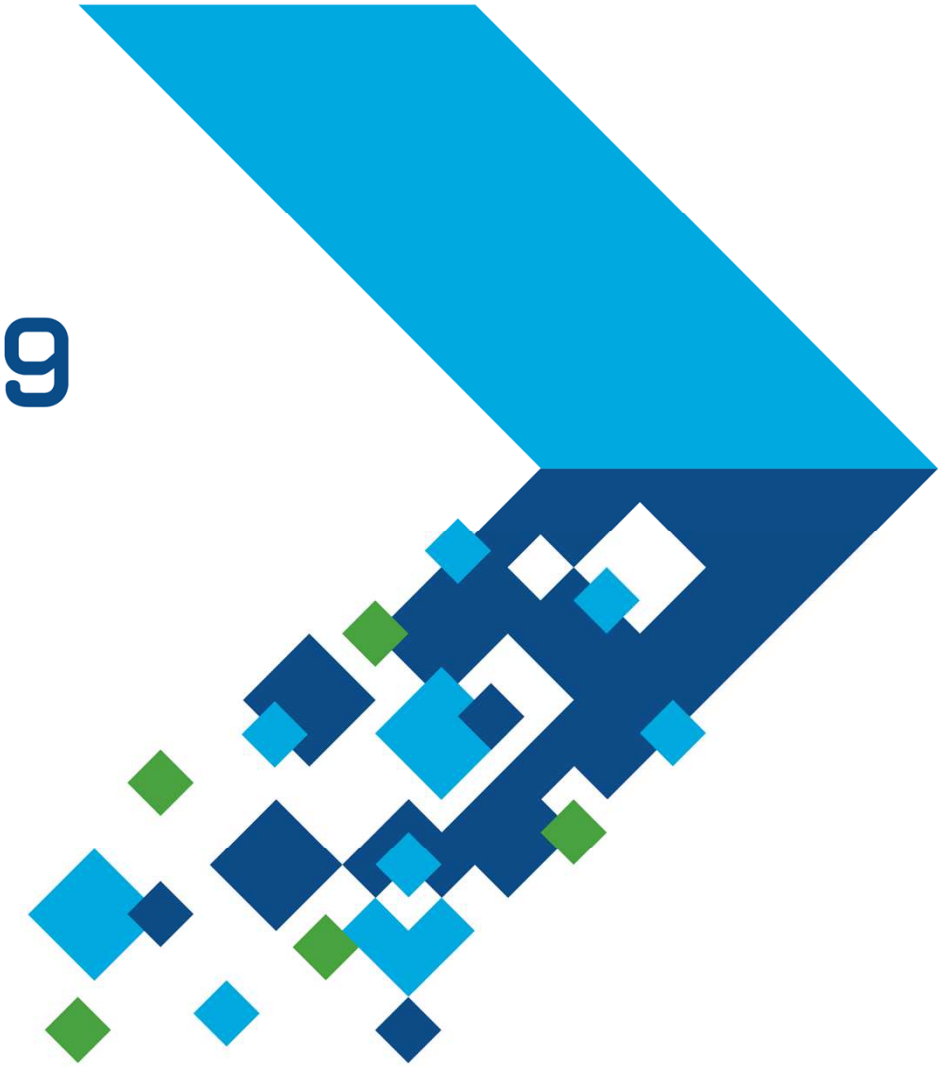
MATLAB EXPO 2019

Systems Engineering

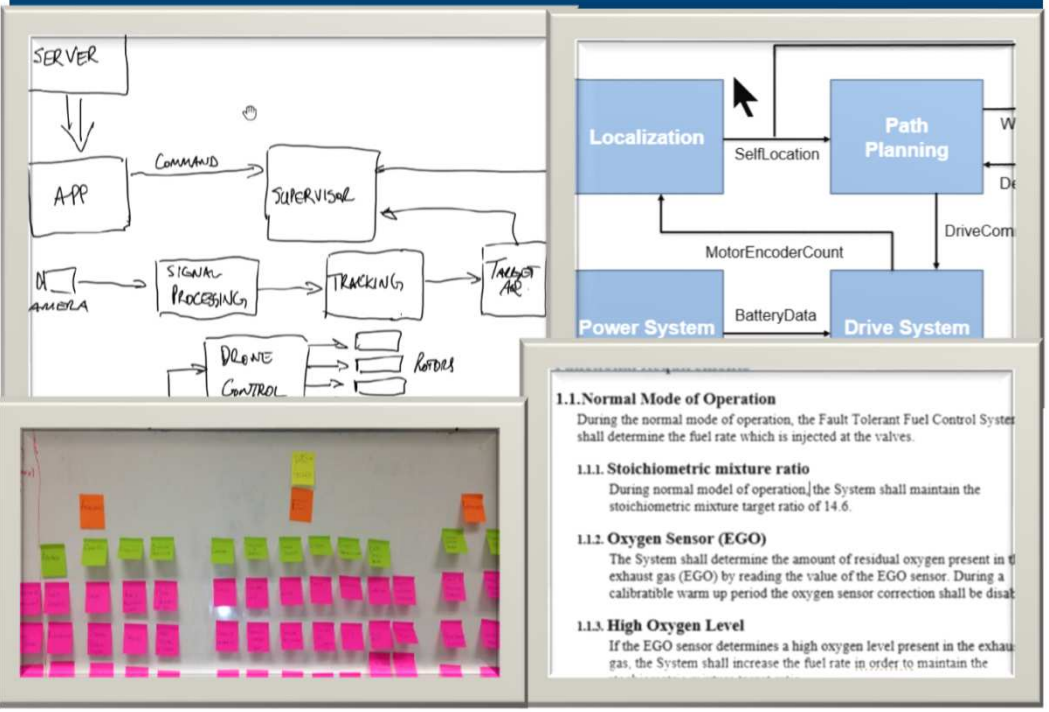
Requirements to Architecture to Simulation

Laurent Royer

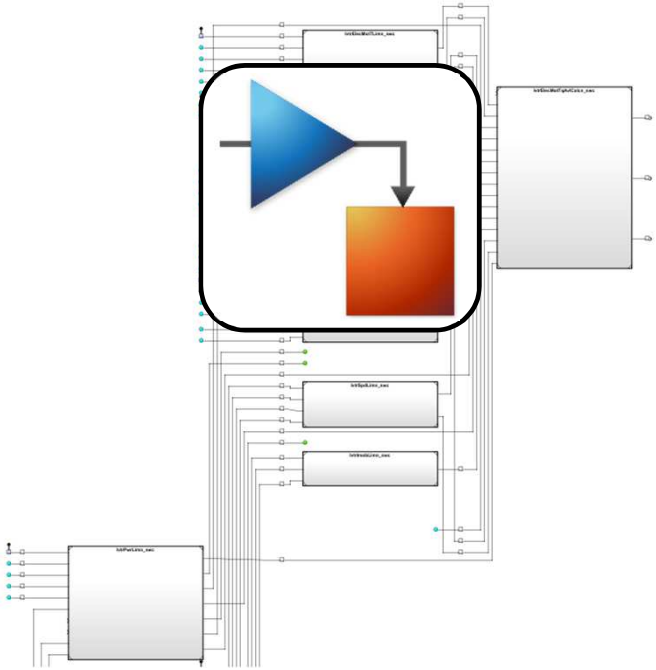


What does the typical process involve?

Early in the Process
Concepts/Descriptions

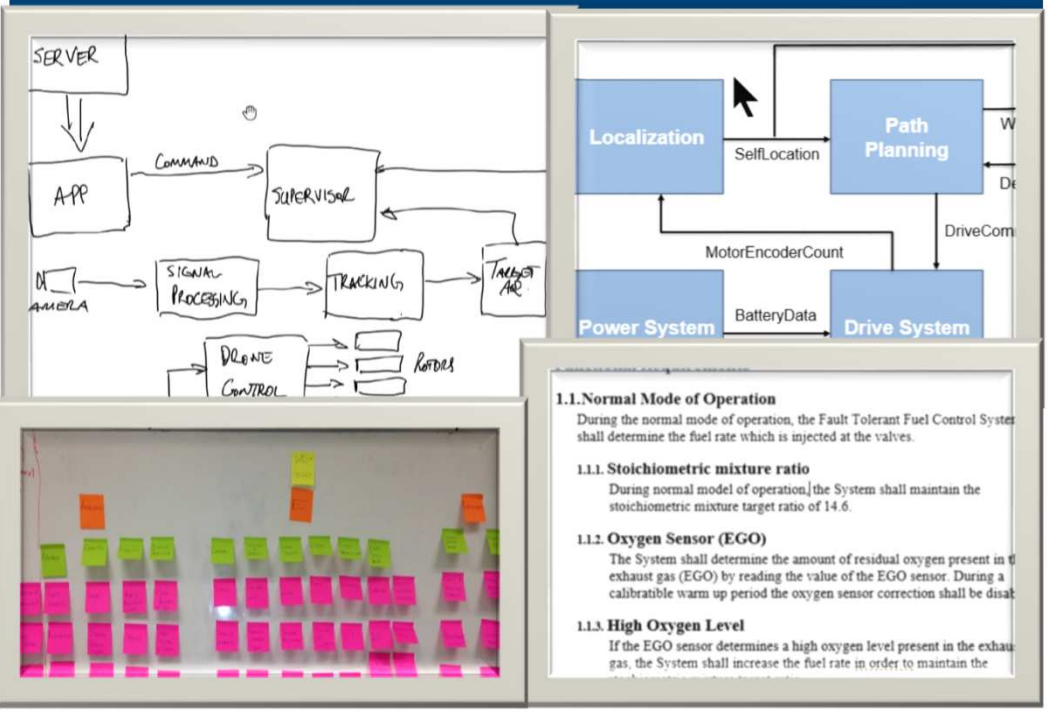


Later in the Process
Models



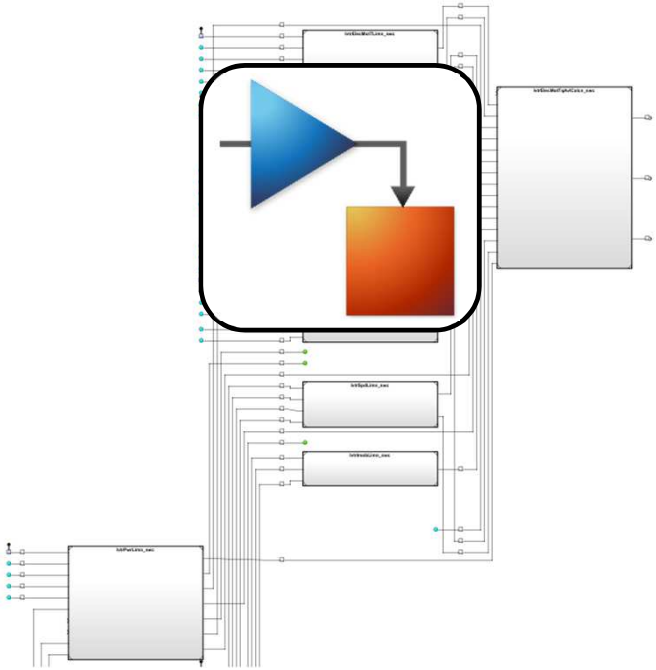
What is the Gap?

Early in the Process
Concepts/Descriptions



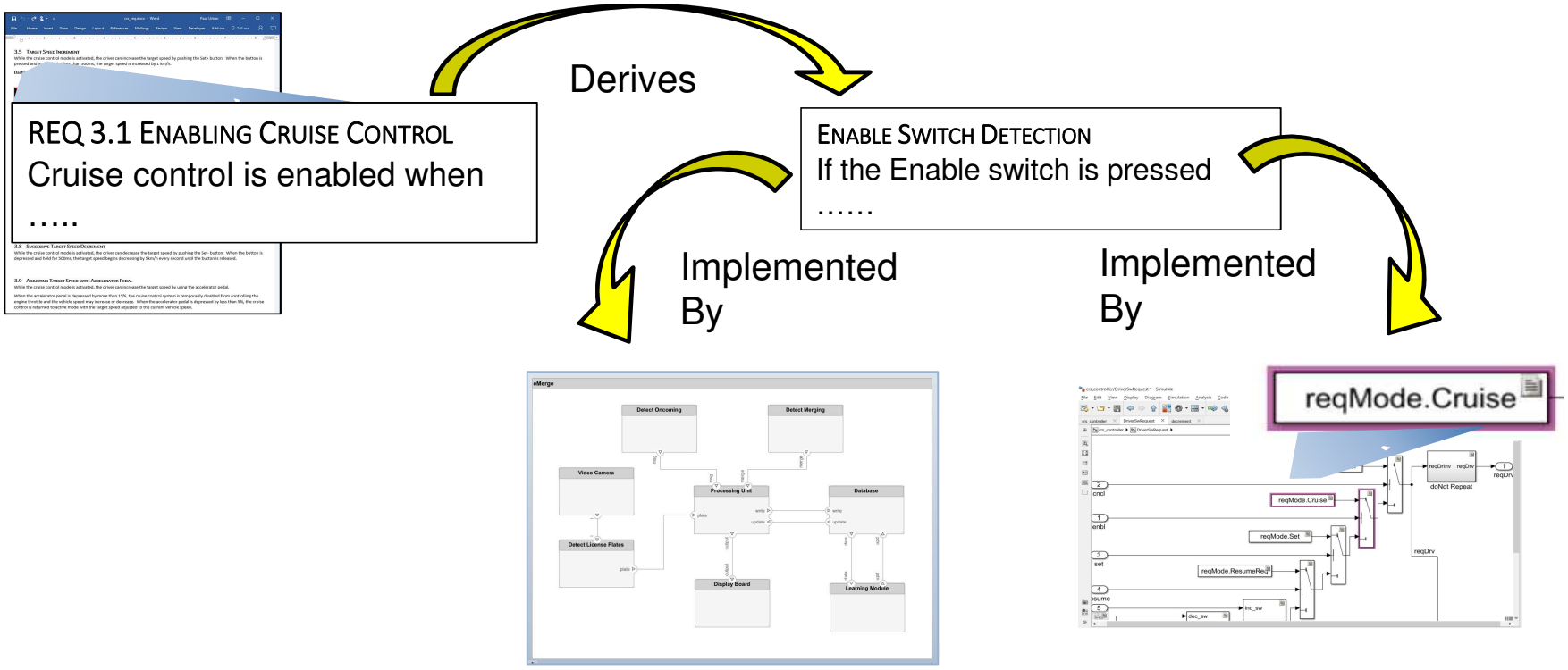
Traceability
Synchronization
Analysis & Simulation

Later in the Process
Models



Key Takeaways

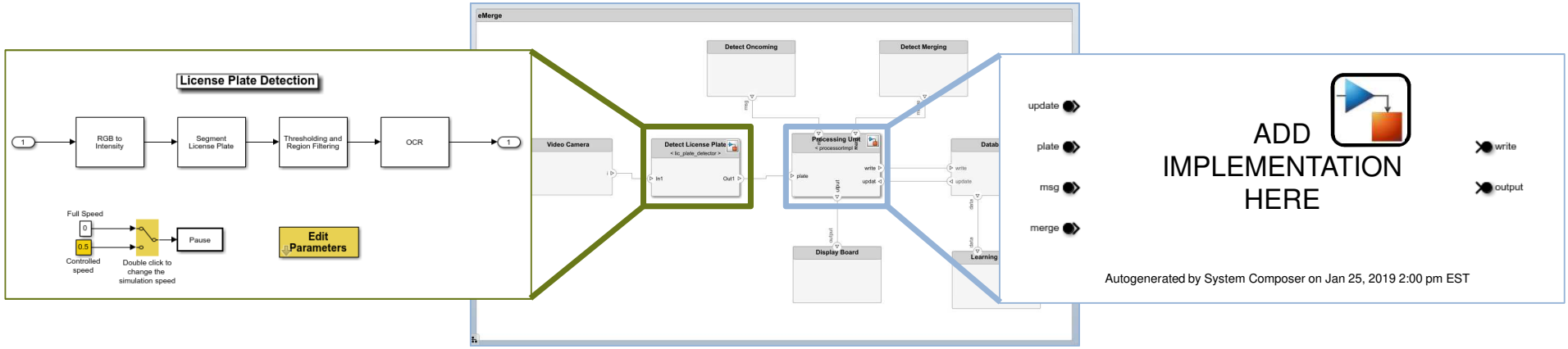
- Digital thread providing traceability between requirements, architecture, and design



Key Takeaways

- Digital thread providing traceability between requirements, architecture, and design

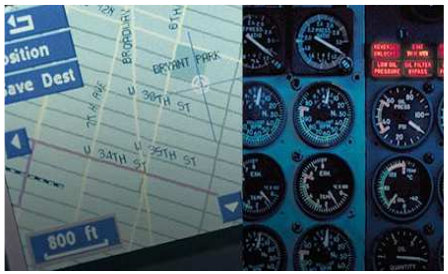
- Connected environment for designing and analyzing architectures and designs



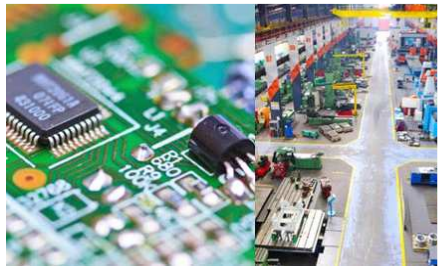
Key Takeaways

- Digital thread providing traceability between requirements, architecture, and design
- Connected environment for designing and analyzing architectures and designs
- Integrated platform for analyzing all parts of your architecture in one multi-domain environment

Dynamic Systems



State Machines



Discrete-Event

Physical Modeling

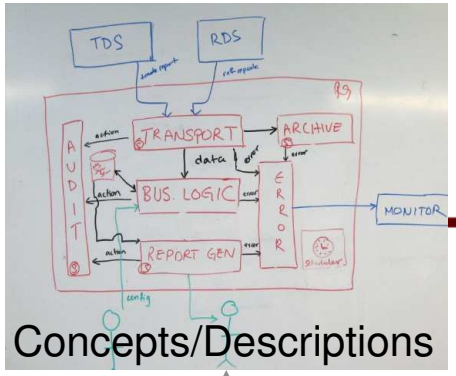
What goes into the bridge?

Be Intuitive

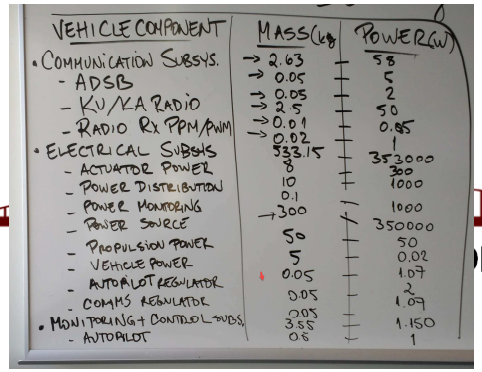
Facilitate Analysis

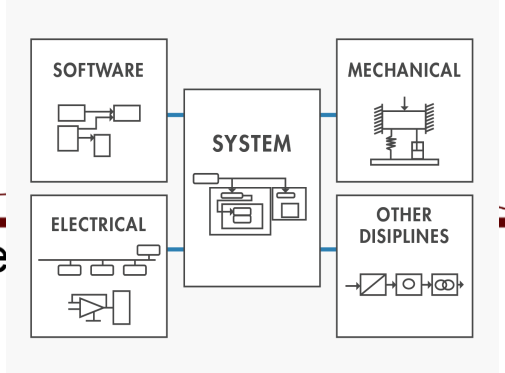
Tackle Complexity

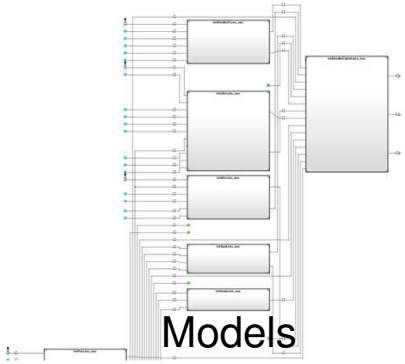
Enable Implementation



Concepts/Descriptions

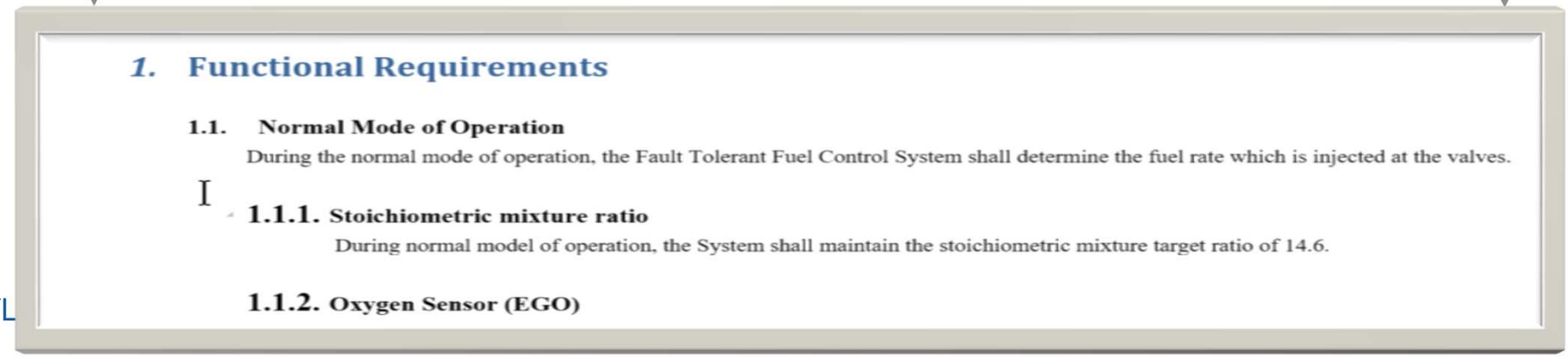






Models

Digital Thread for Traceability



1. Functional Requirements

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.

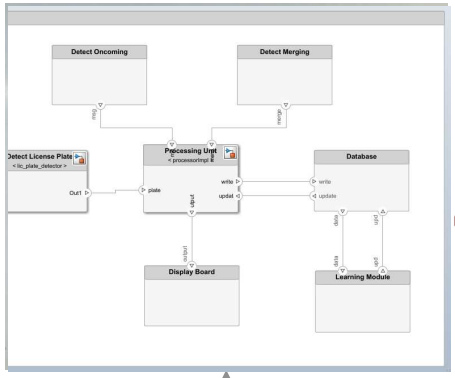
1.1.1. Stoichiometric mixture ratio

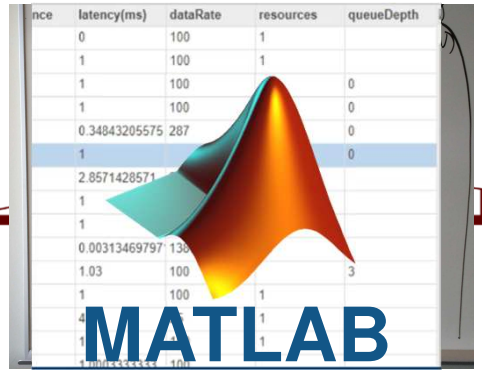
During normal model of operation, the System shall maintain the stoichiometric mixture target ratio of 14.6.

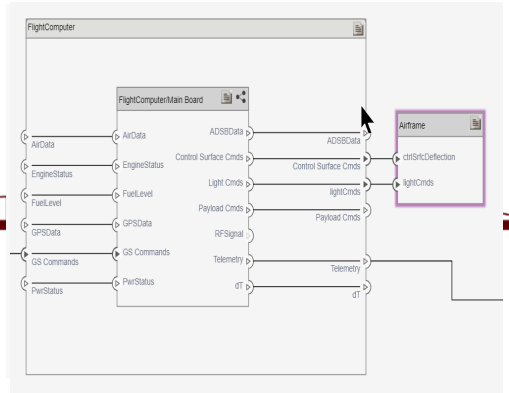
1.1.2. Oxygen Sensor (EGO)

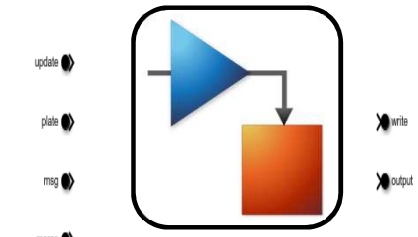
MathWorks Solution: System Composer R2019a

✓Be Intuitive ✓Facilitate Analysis ✓Tackle Complexity ✓Enable Implementation





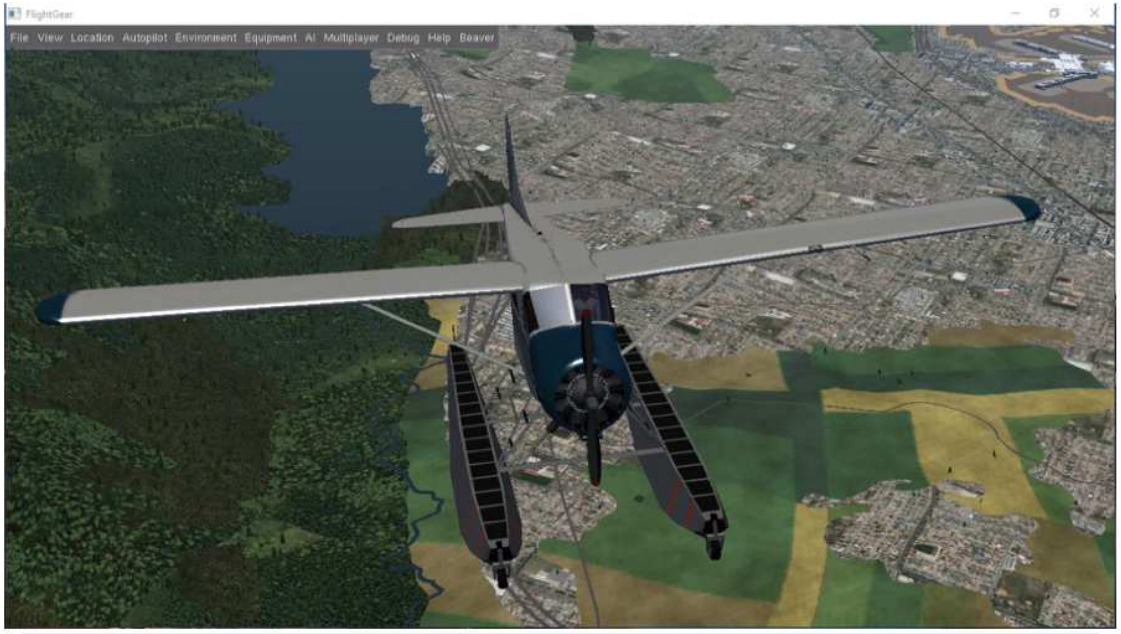




Requirements Coverage Reporting and Impact Analysis

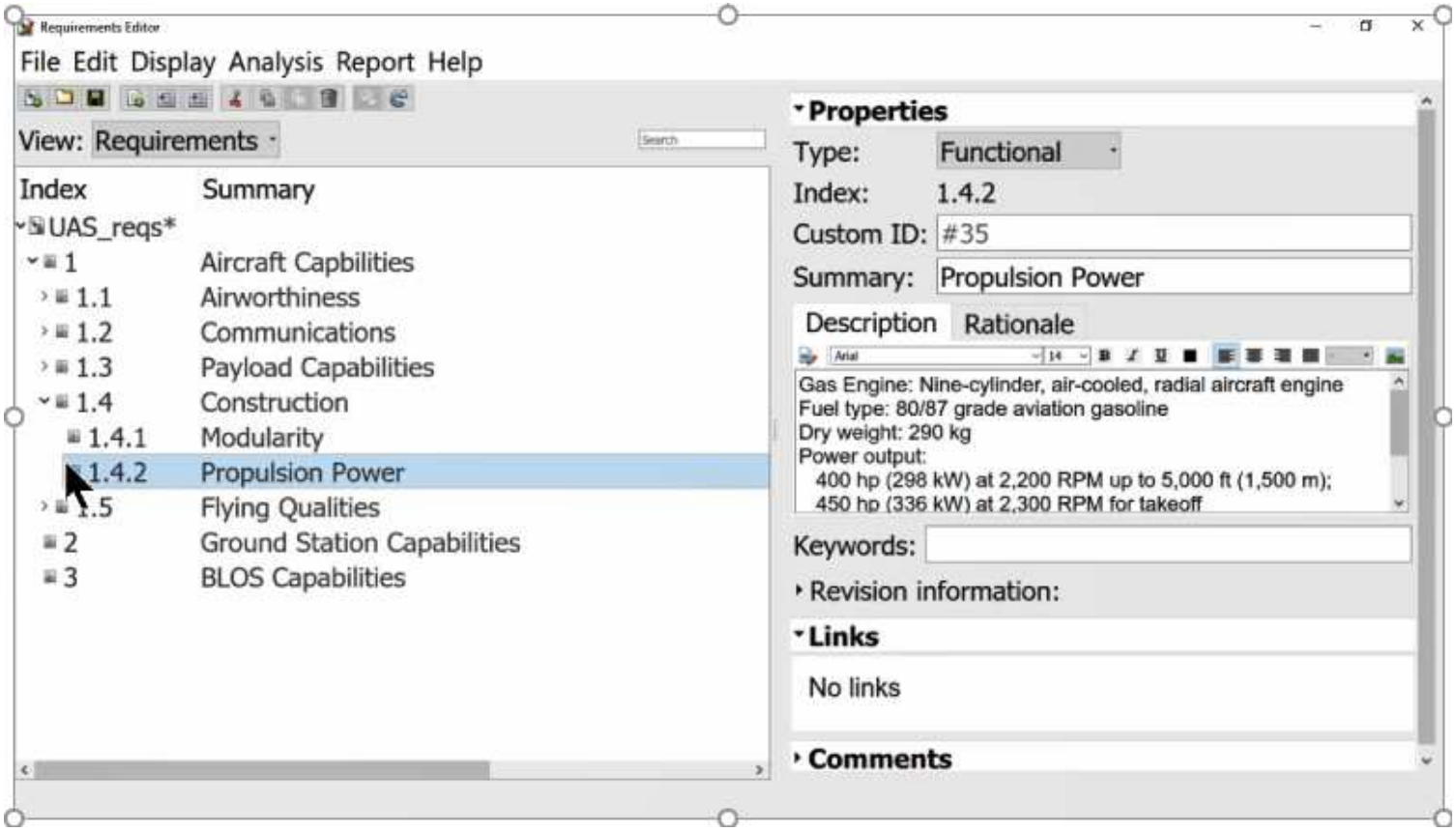
Now let’s see it in action

De Havilland DHC-2 "Beaver"



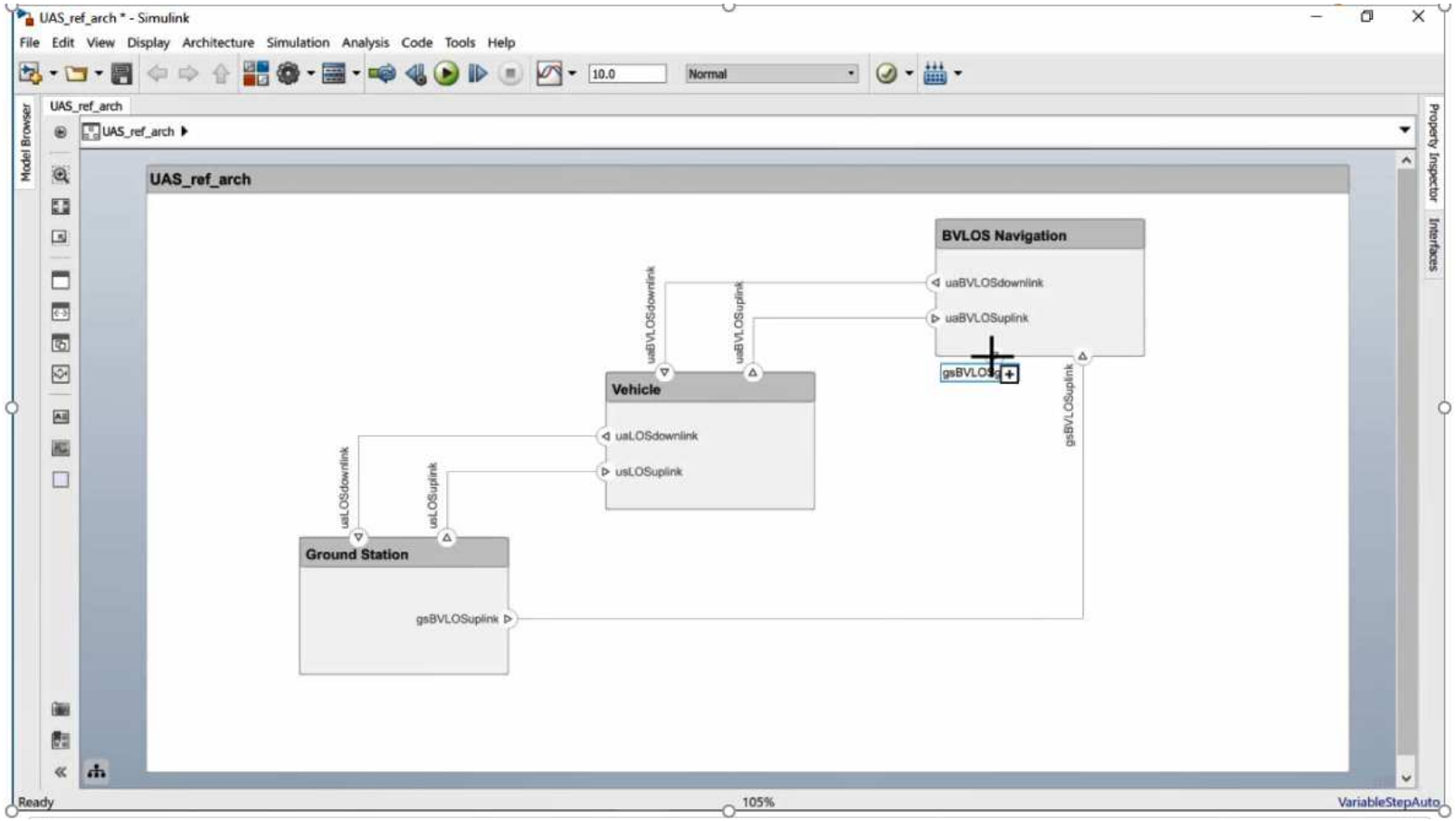
Update the De Havilland "Beaver" airplane to have full electrification of the propulsion system and make it an optionally piloted aircraft

Requirements



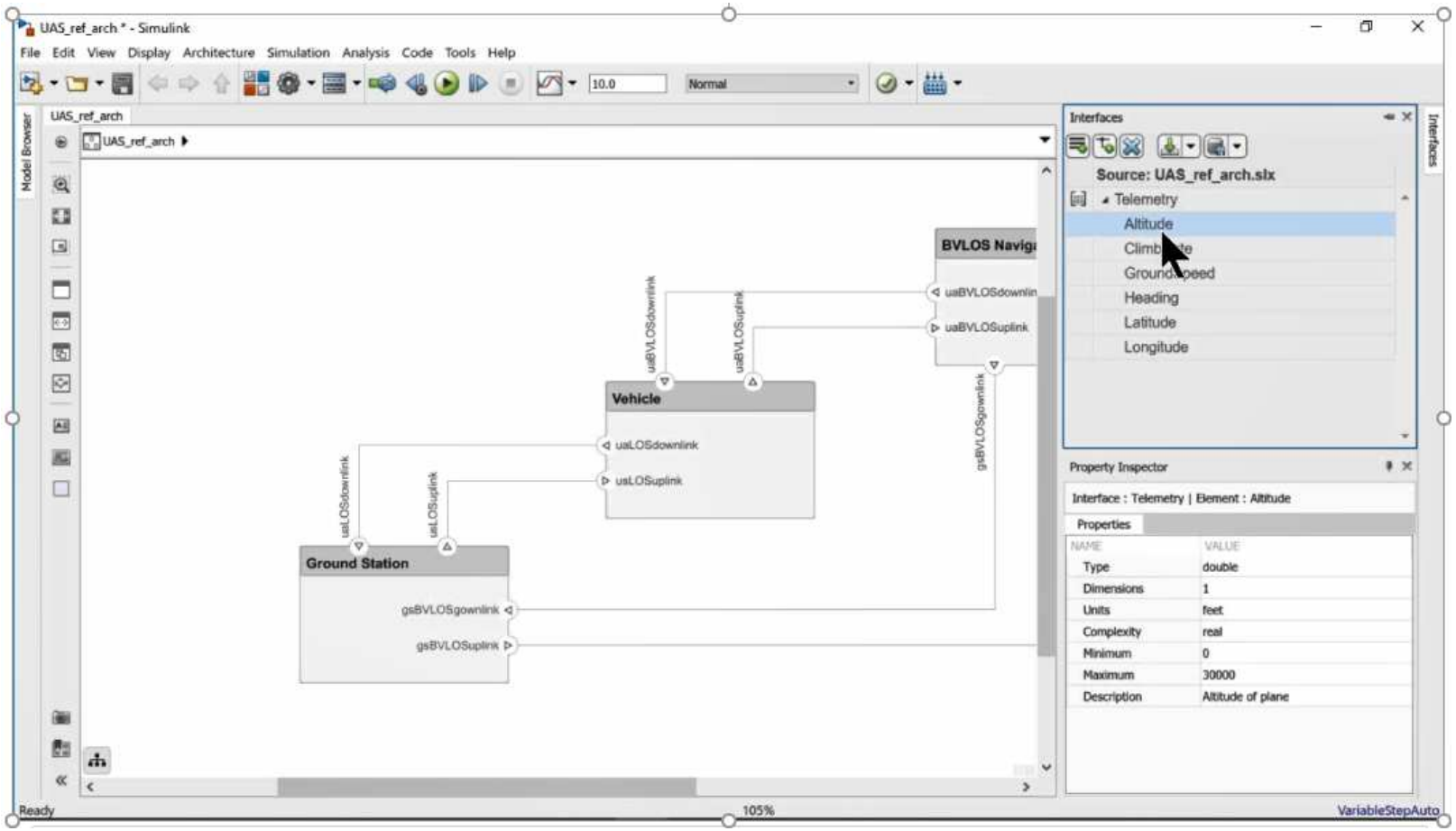
Requirements imported from Word and captured in Simulink Requirements

Components

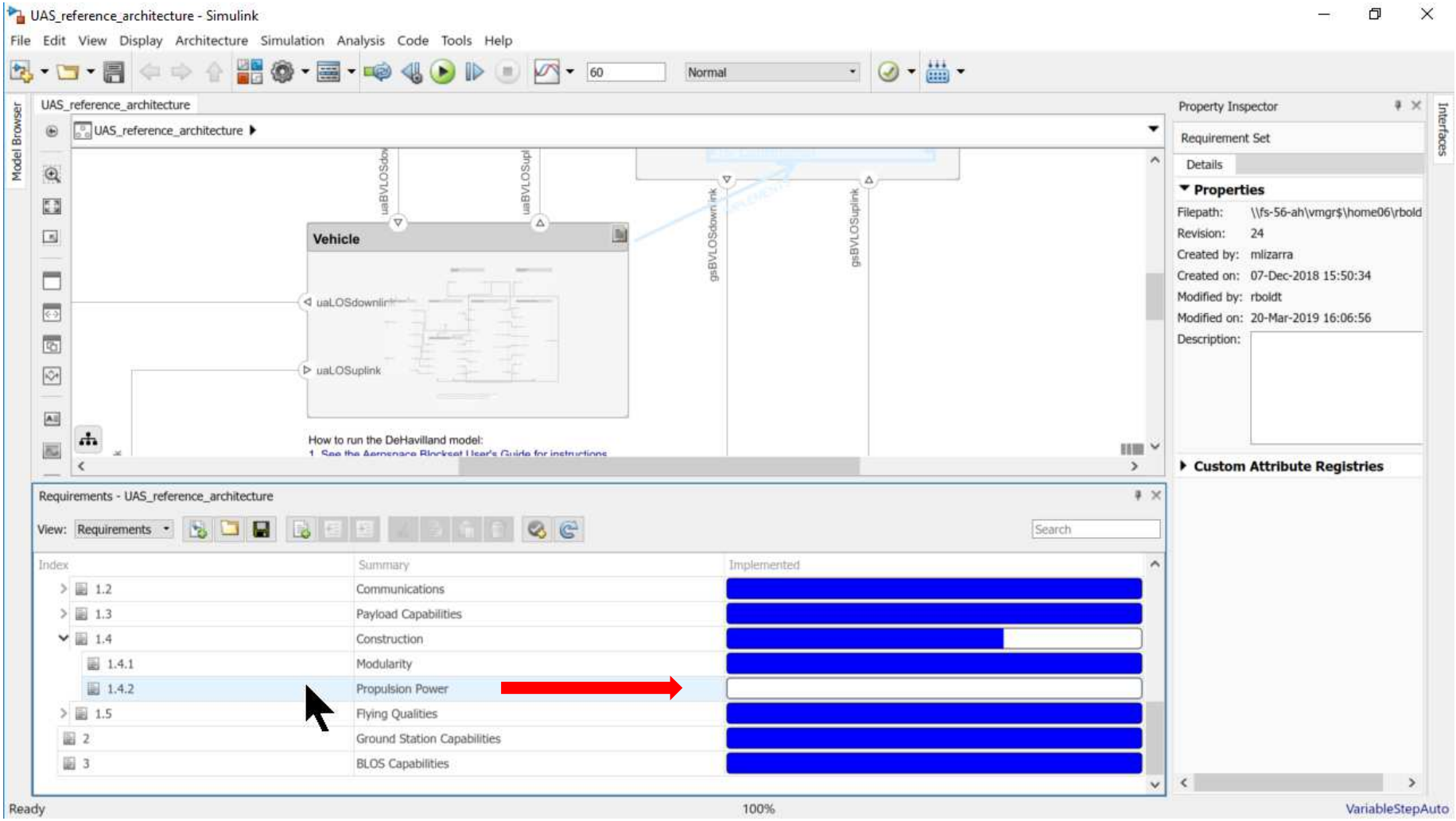


Create components for a ground station and beyond-visual-line-of-sight navigation in addition to the vehicle

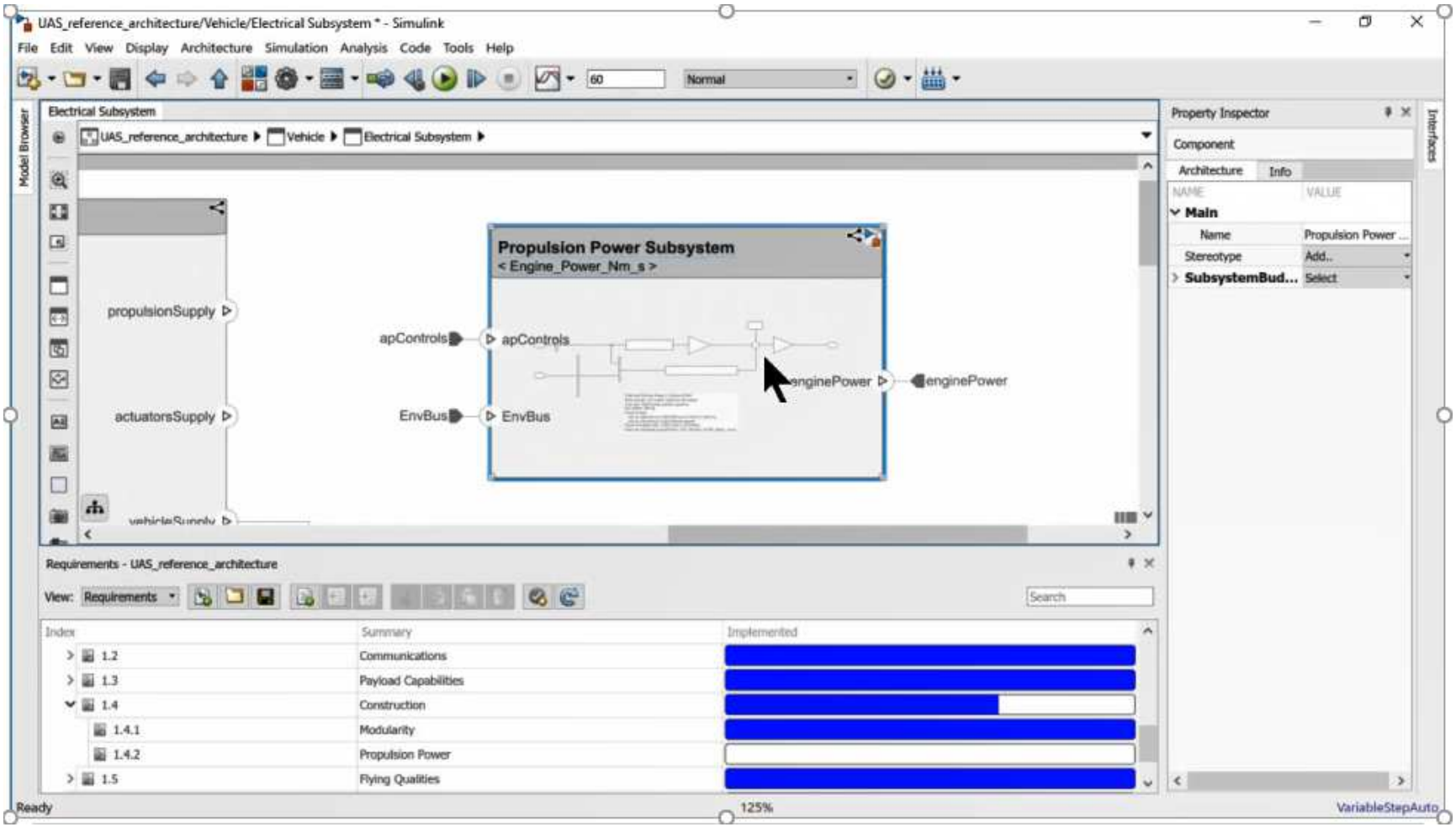
Interfaces



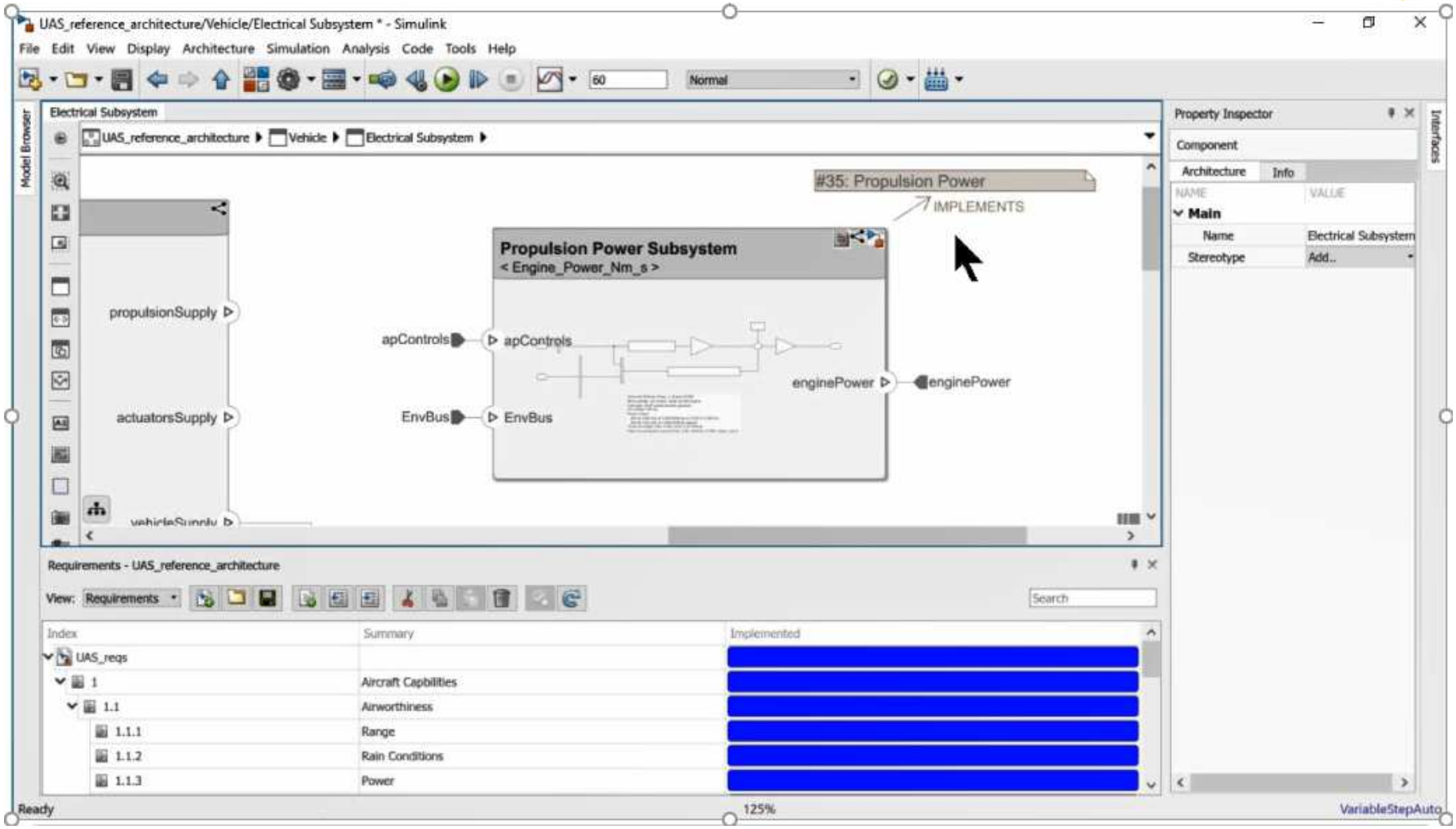
Add details by defining interfaces



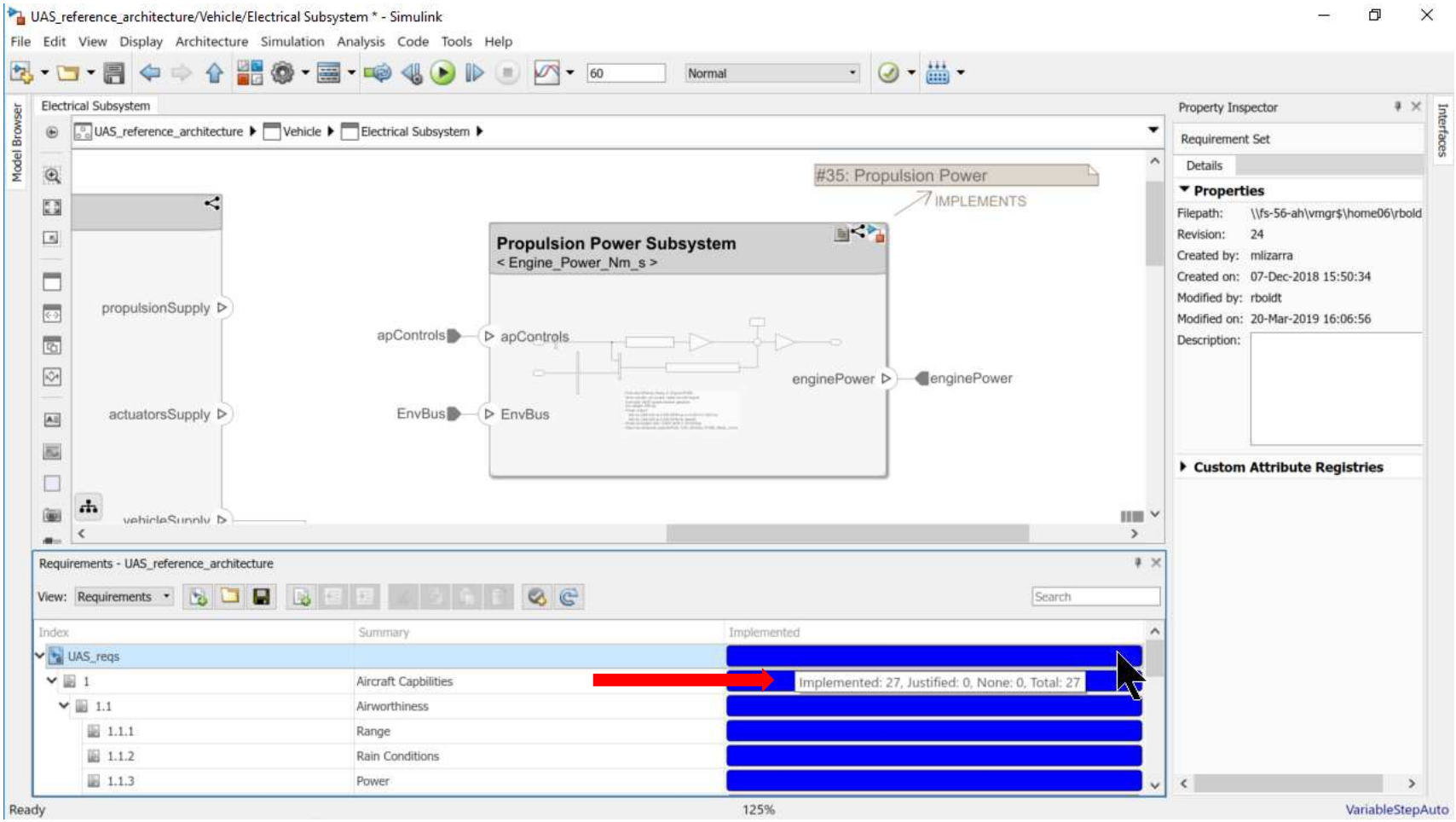
Every requirement except Propulsion Power has been implemented



Drilling down in our component hierarchy we can find the Propulsion Power Subsystem Component

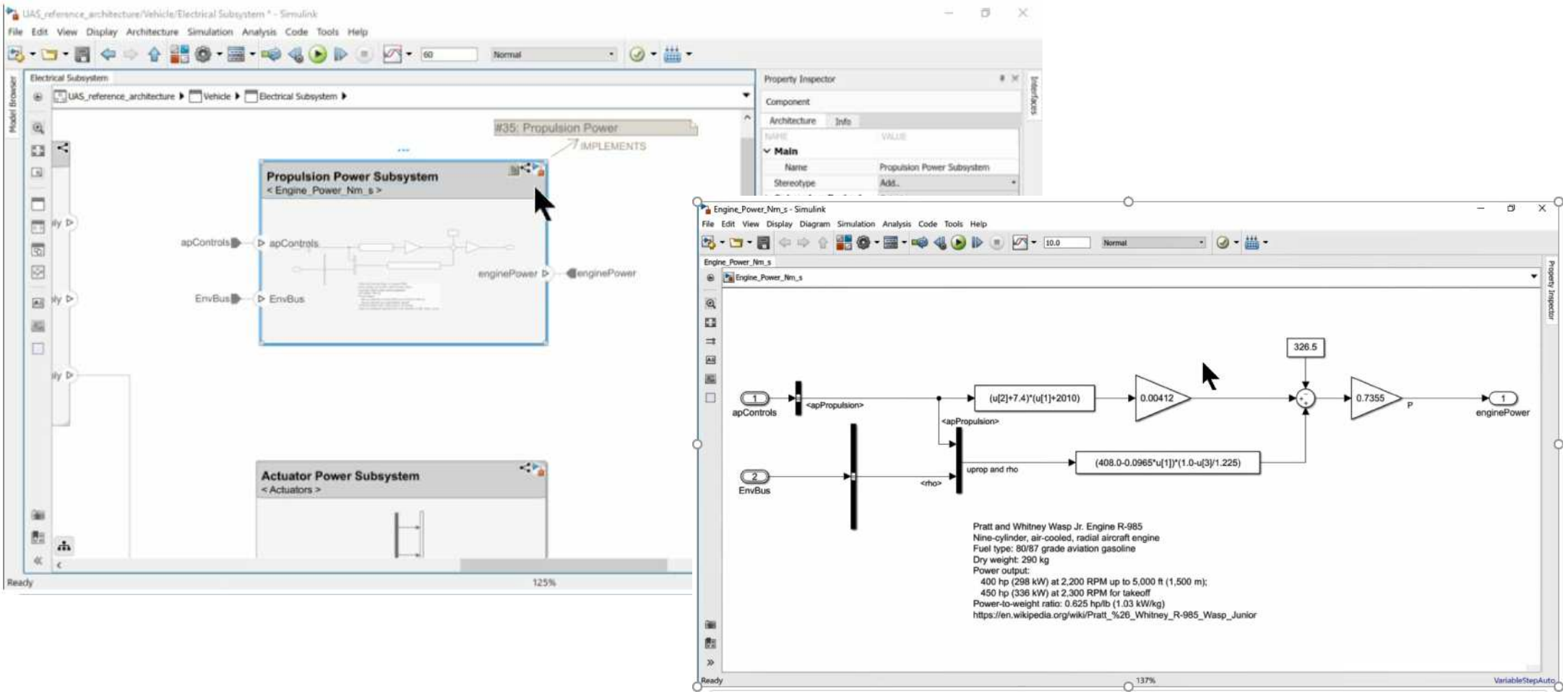


We associate this component to the requirement on Propulsion Power to specify how this requirement is implemented

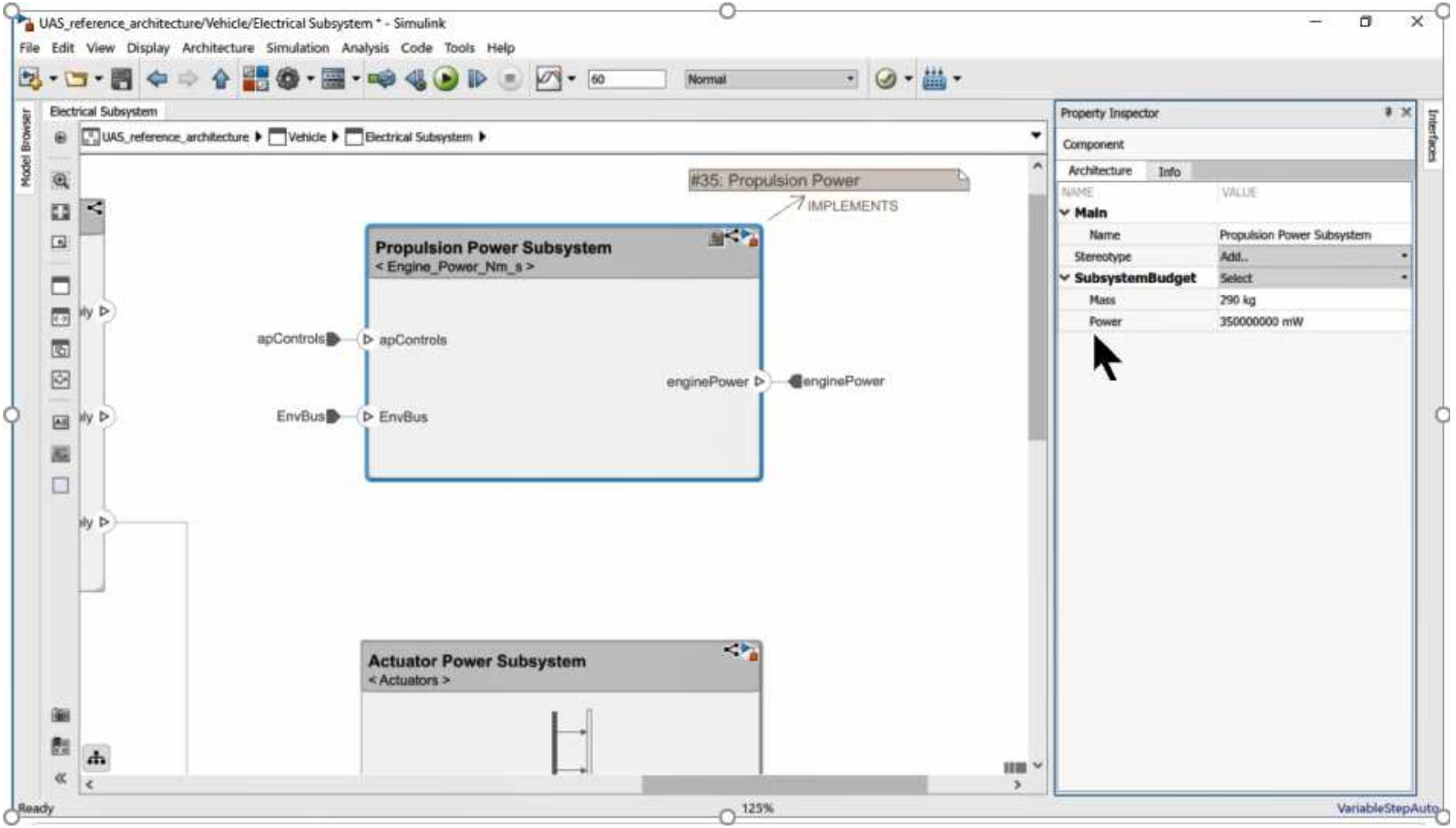


We now have implemented every requirement

Simulink badge

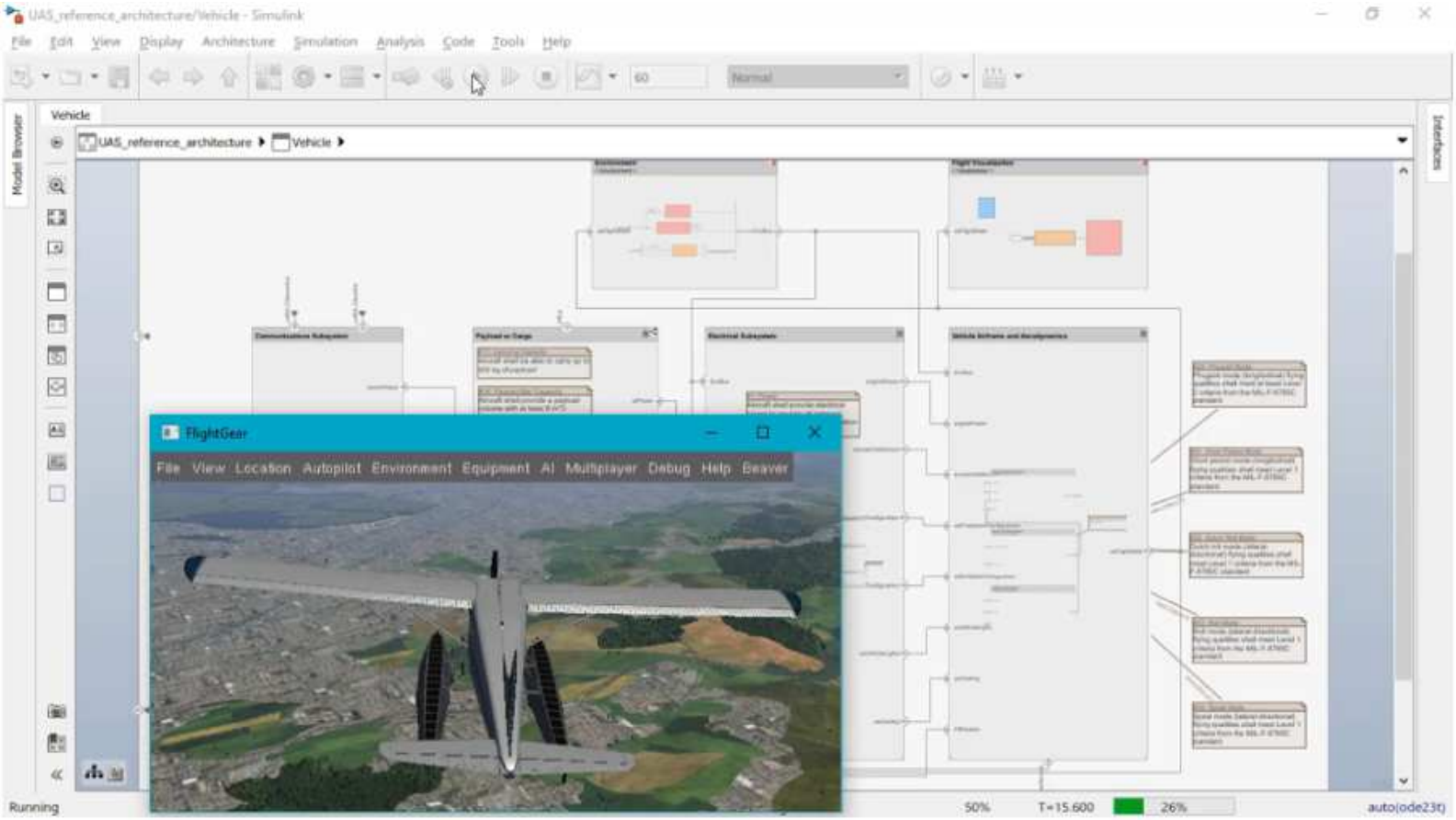


The Simulink badge tells us that this component is linked to the Engine Power Simulink model

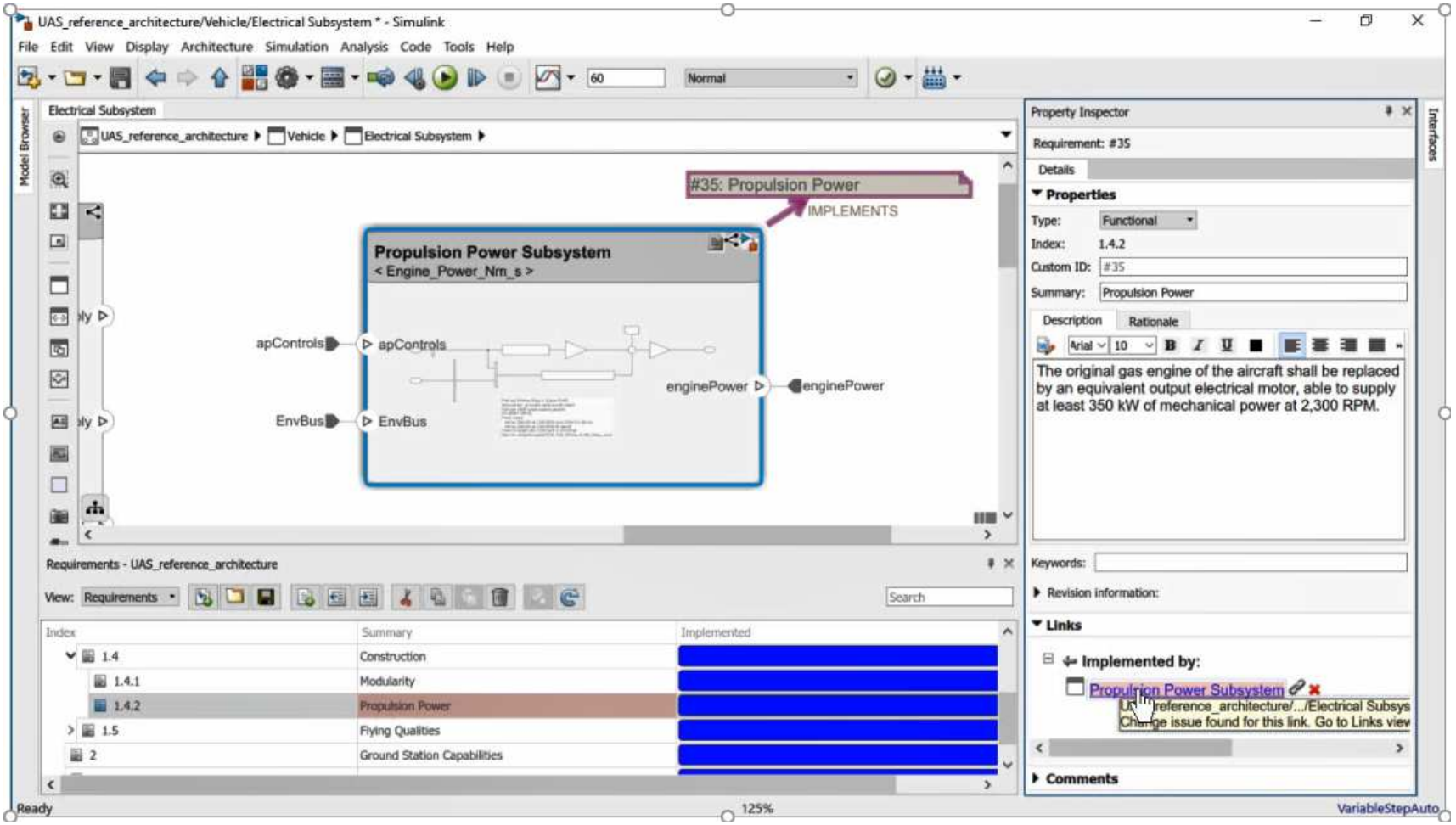


User-defined properties and their values can be accessed by MATLAB to run various types of analysis

Electrification upgrade

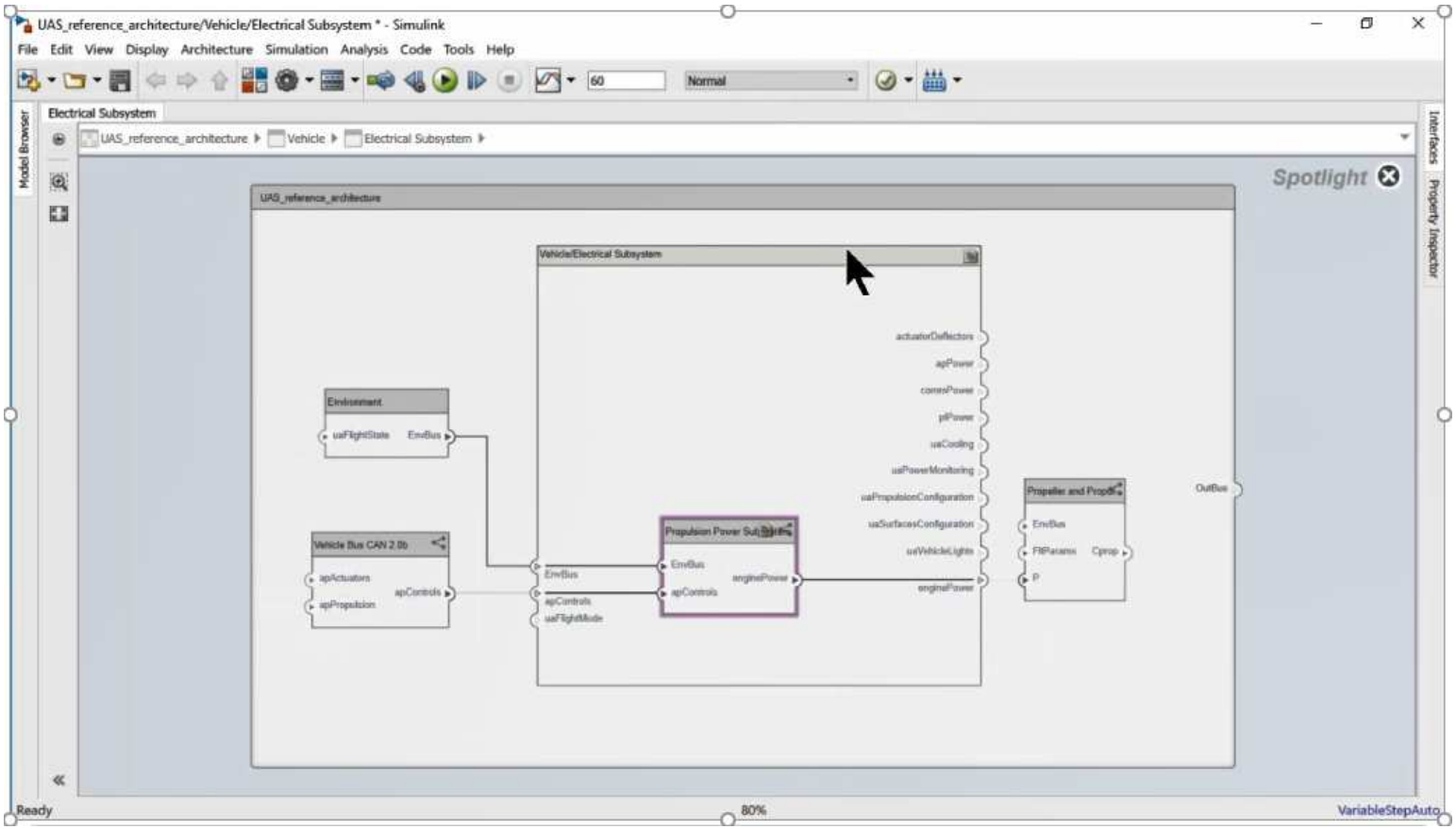


Simulation is OK, move on to the electrification upgrade

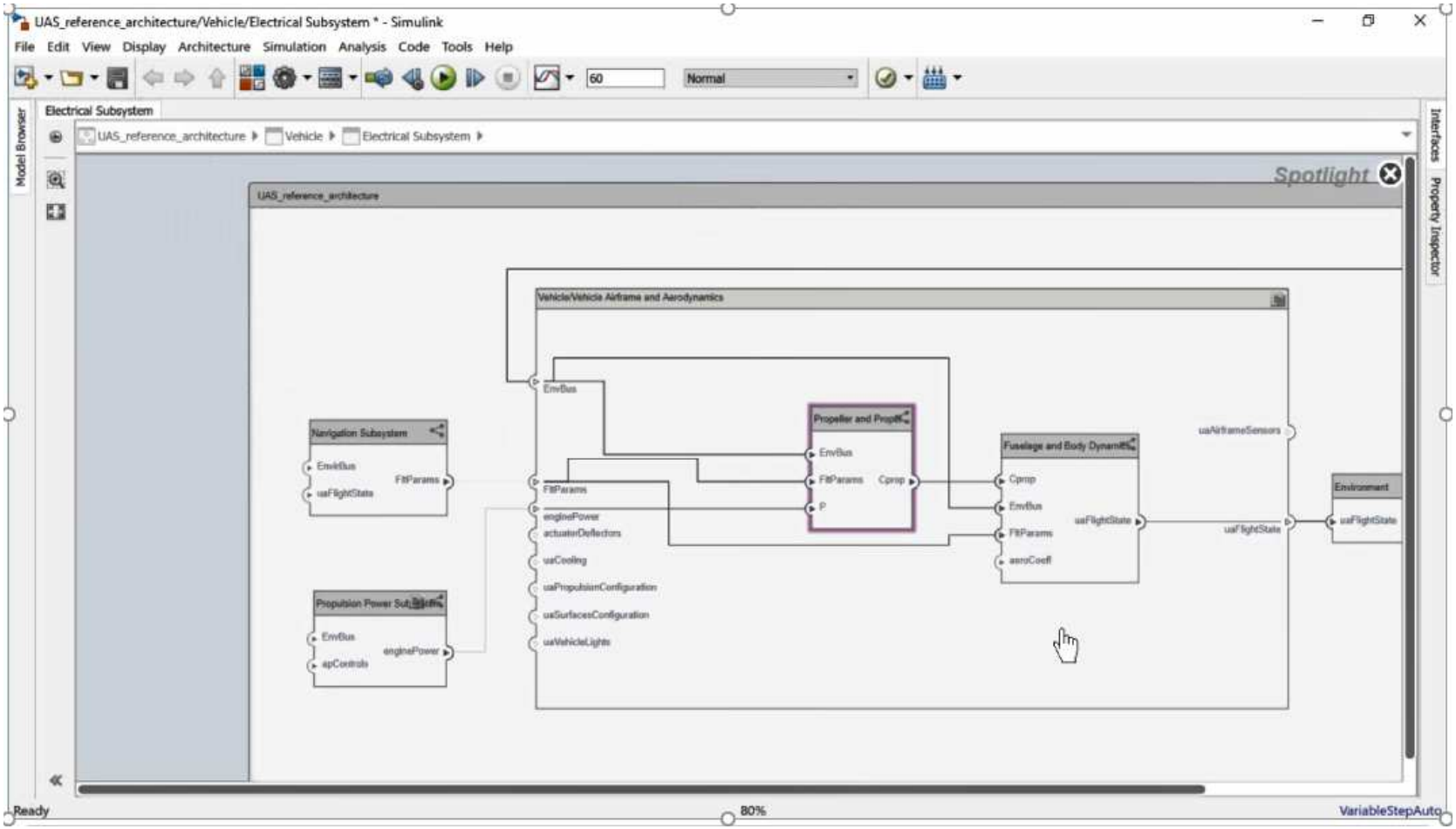


We now change the Propulsion Power requirement to specify that the power must come from an electric motor

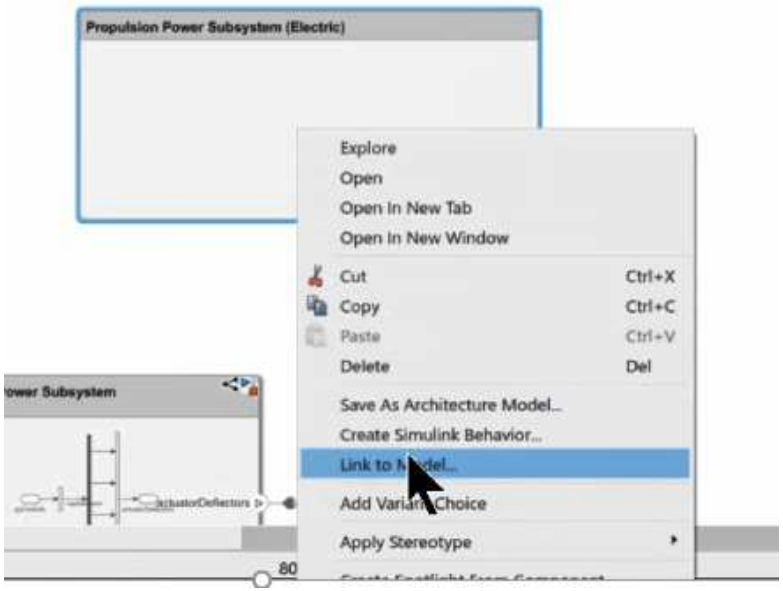
Spotlight view

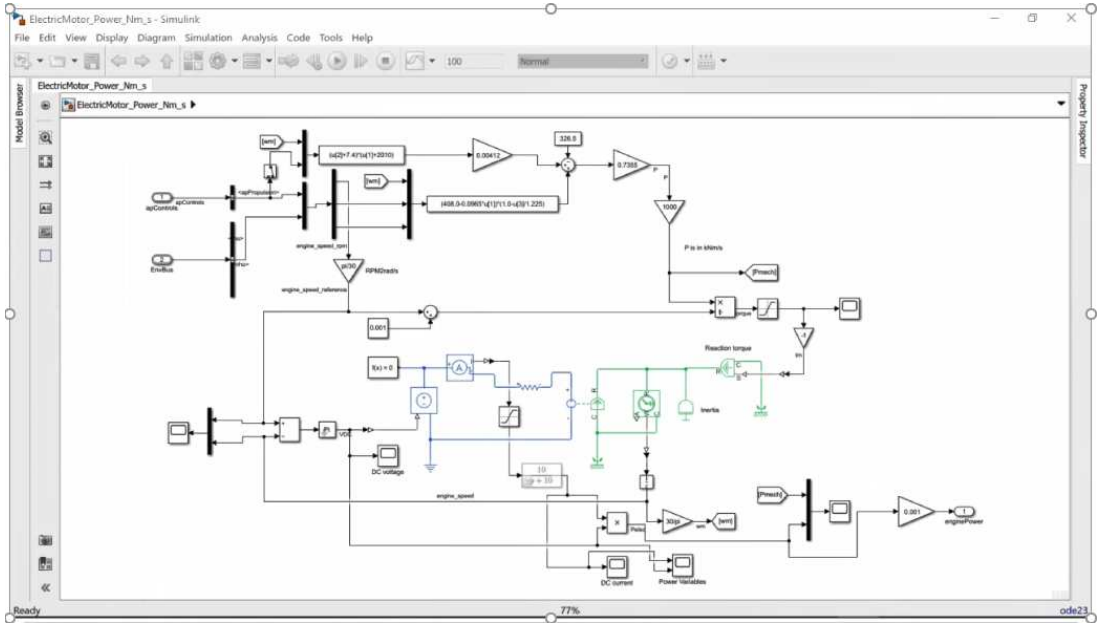


The spotlight view shows what other components may be directly impacted by the change

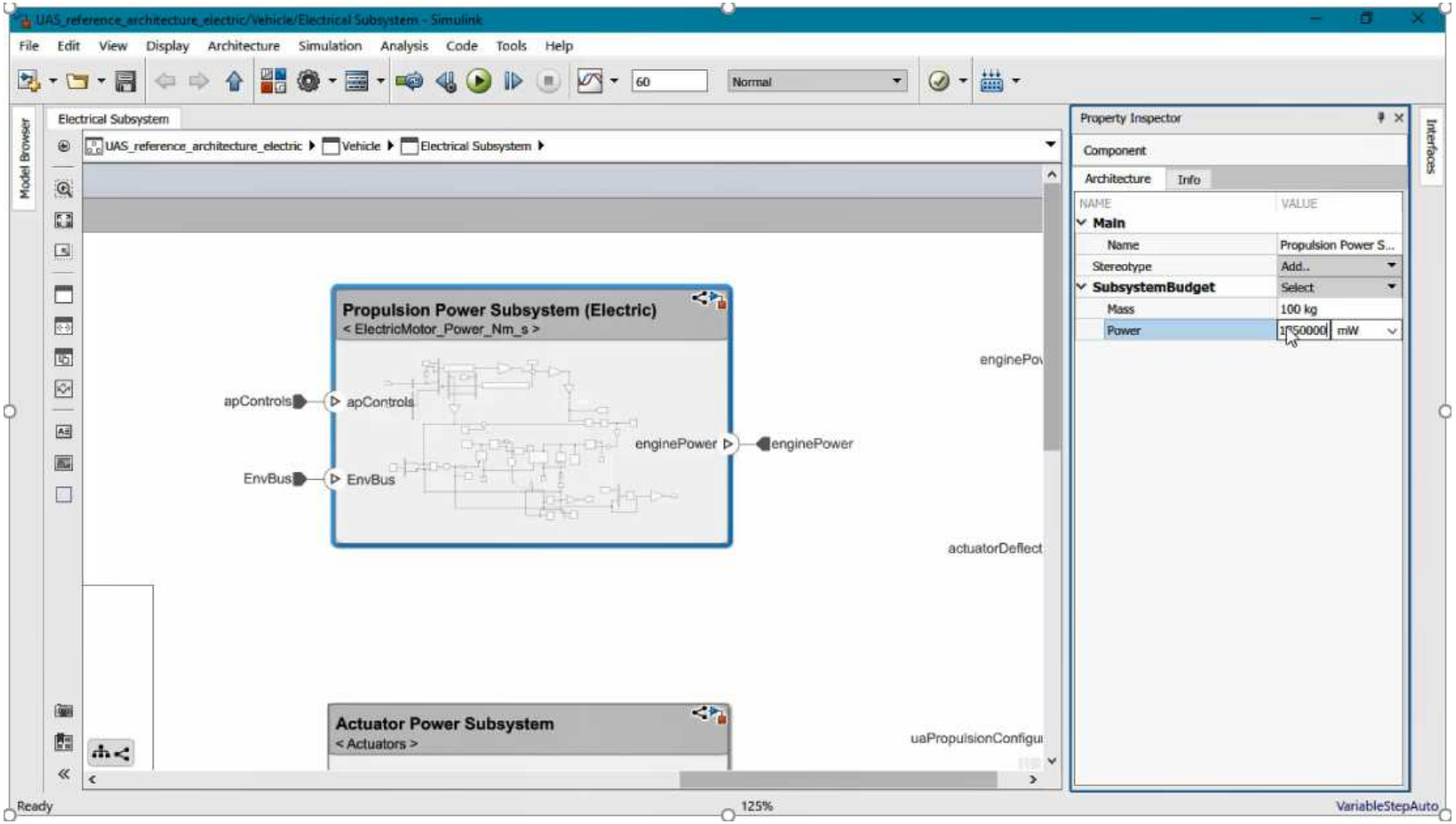


The effect this may have on other components can be followed further downstream by creating a spotlight of the Propeller and Propulsion component

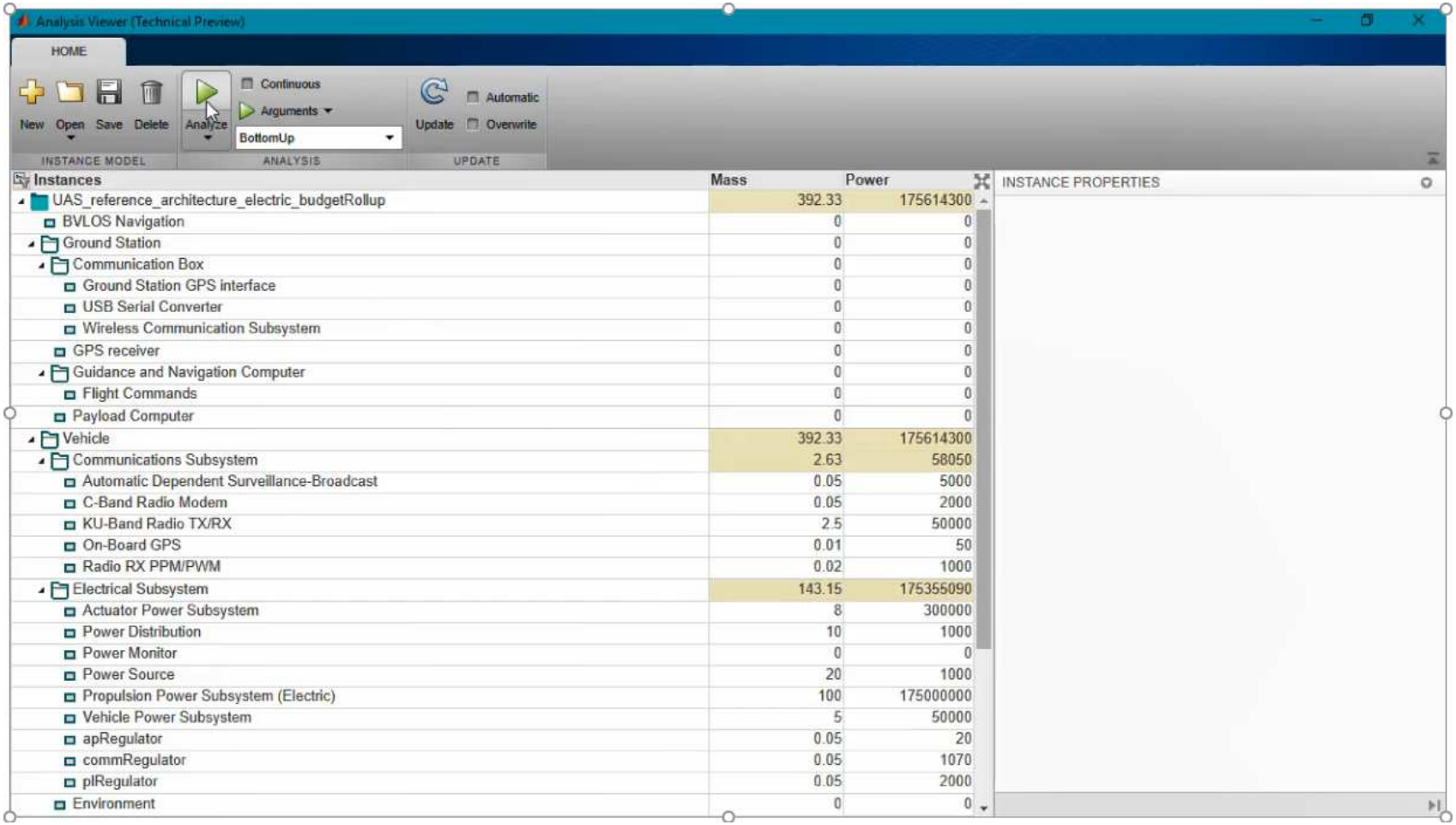




We replace the current Propulsion Power component with a new one with an electric motor Simulink model linked to it

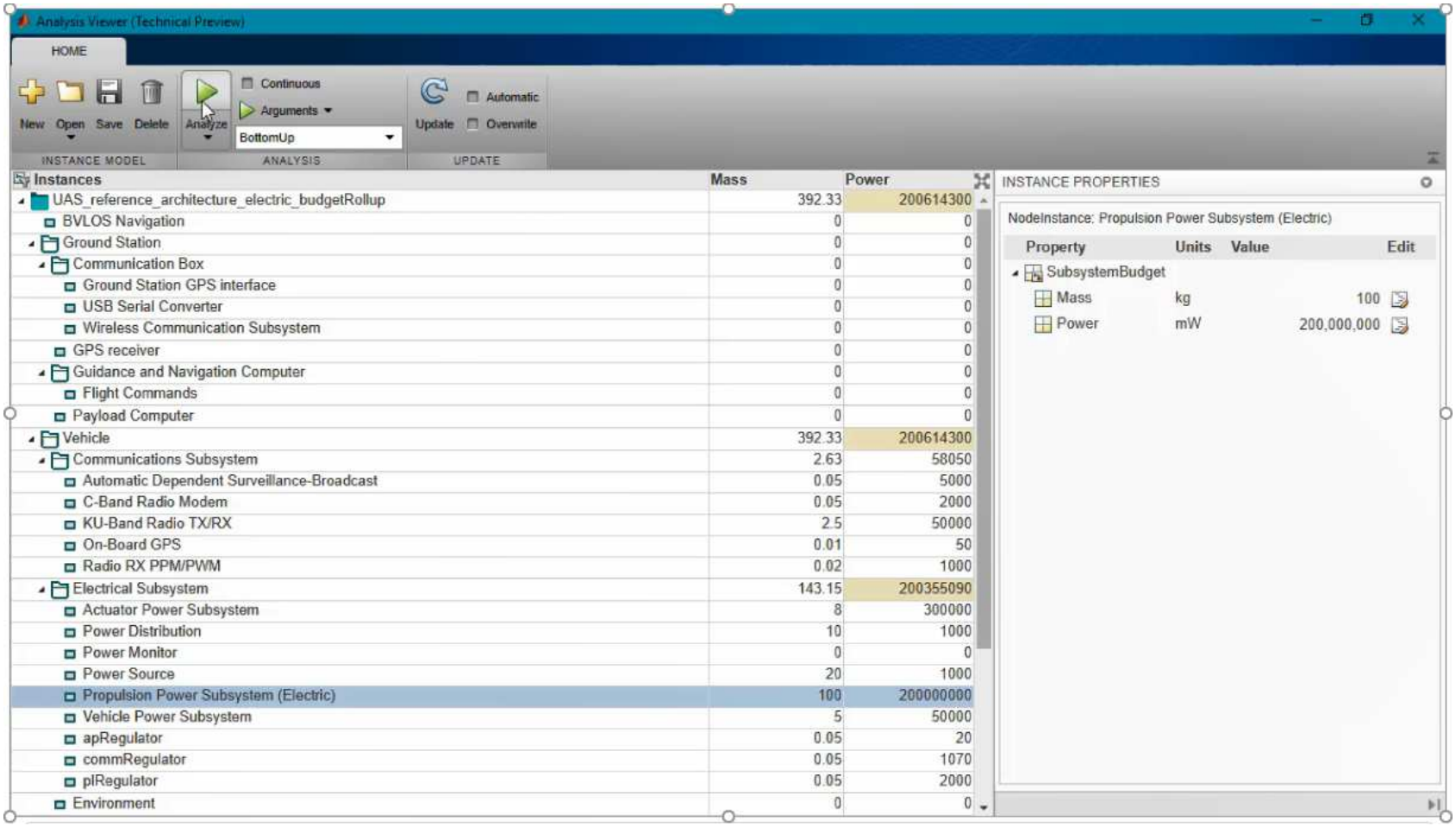


To see how this change impacts the total mass and power of the plane, we first add the value of mass and power to the new component by applying a stereotype to the component, and set the values of mass and power properties.

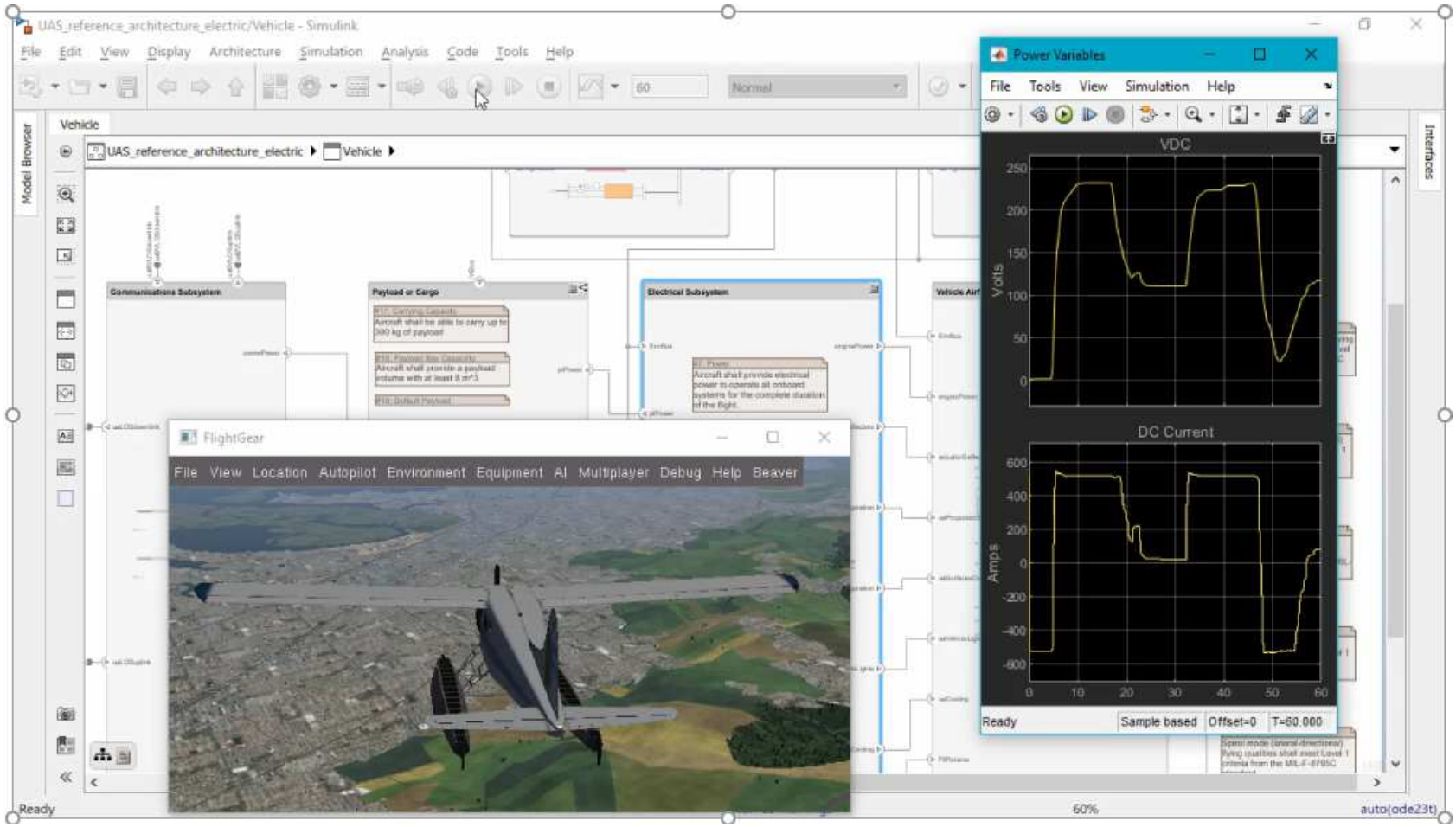


Instances	Mass	Power
UAS_reference_architecture_electric_budgetRollup	392.33	175614300
BVLOS Navigation	0	0
Ground Station	0	0
Communication Box	0	0
Ground Station GPS interface	0	0
USB Serial Converter	0	0
Wireless Communication Subsystem	0	0
GPS receiver	0	0
Guidance and Navigation Computer	0	0
Flight Commands	0	0
Payload Computer	0	0
Vehicle	392.33	175614300
Communications Subsystem	2.63	58050
Automatic Dependent Surveillance-Broadcast	0.05	5000
C-Band Radio Modem	0.05	2000
KU-Band Radio TX/RX	2.5	50000
On-Board GPS	0.01	50
Radio RX PPM/PWM	0.02	1000
Electrical Subsystem	143.15	175355090
Actuator Power Subsystem	8	300000
Power Distribution	10	1000
Power Monitor	0	0
Power Source	20	1000
Propulsion Power Subsystem (Electric)	100	175000000
Vehicle Power Subsystem	5	50000
apRegulator	0.05	20
commRegulator	0.05	1070
plRegulator	0.05	2000
Environment	0	0

Using the Analysis model, we can create an analysis matrix that can be used to perform trade studies. These trade studies will be used to guide us in optimizing the architecture.



We can change the values of properties to run what-if calculations to do trade-offs of the system

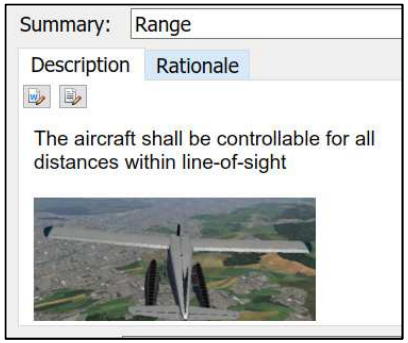


As a next step, running a simulation would allow us to size the battery that will act as the power source for the new electric motor

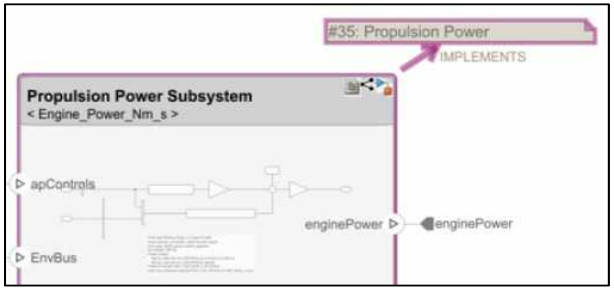
Simulink Requirements

Digital Thread from Requirements to Architecture and Design

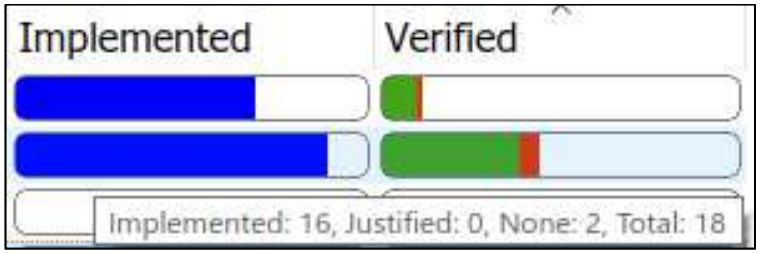
Author requirements or view from external source



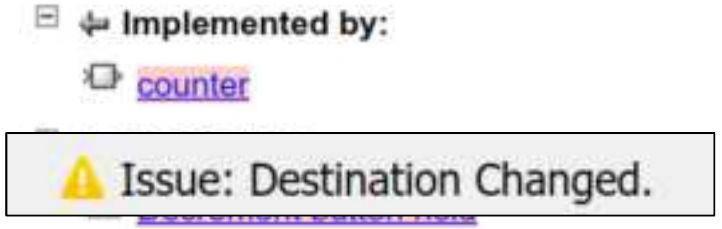
Link requirements, architectures, design, code and test



Identify gaps in architecture or design



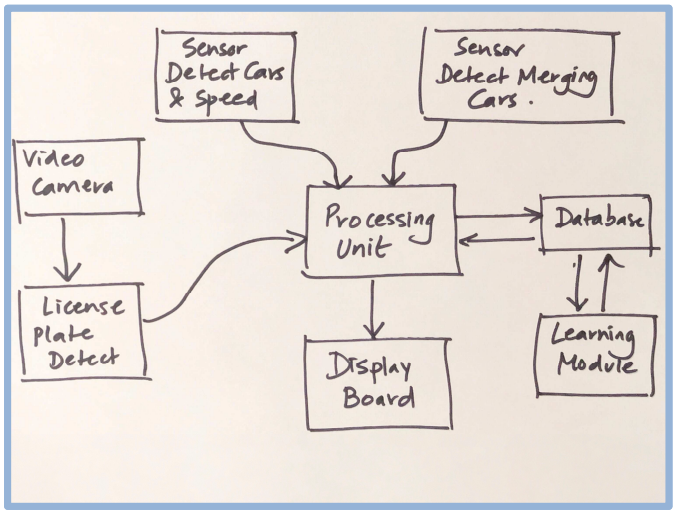
Identify impact of requirement changes



System Composer

Intuitively design system and software architectures

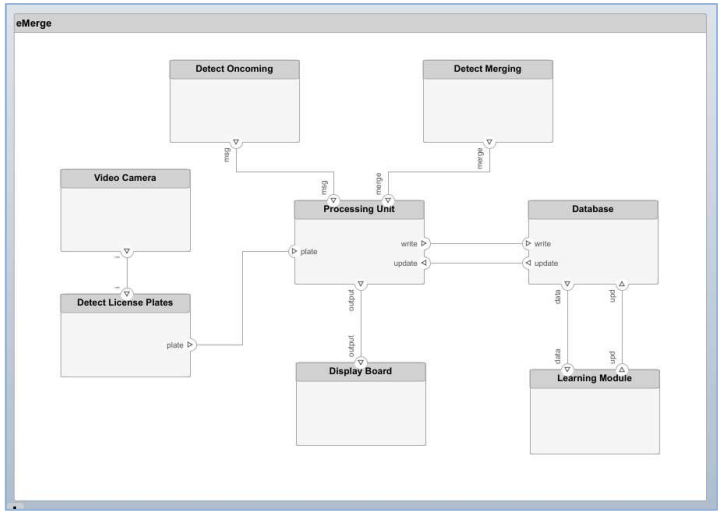
R2019a



Description

==

Architecture

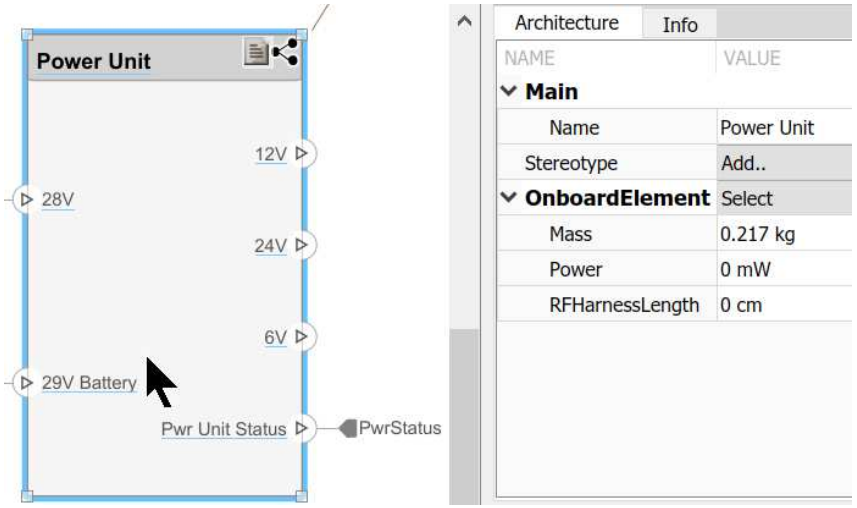


System Composer

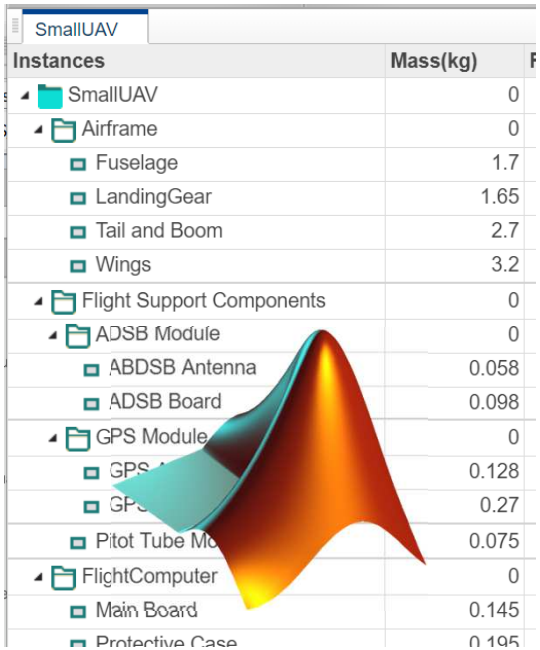
R2019a

Perform trade studies based on data driven analysis to optimize architectures

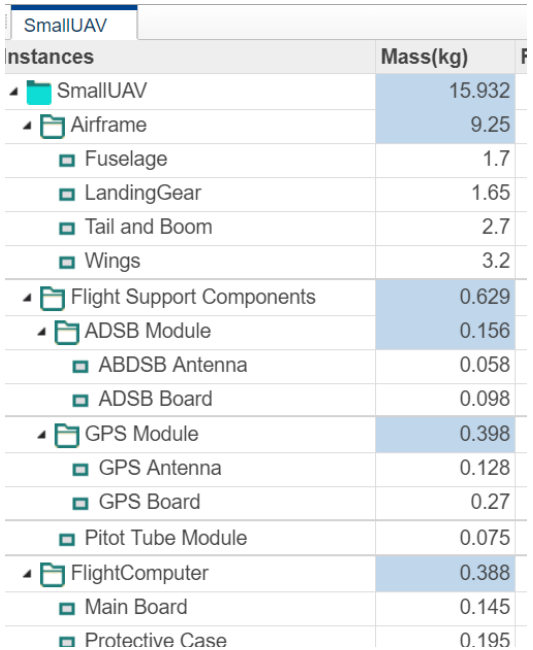
Add custom data



Create analysis model



Calculate mass roll-up data

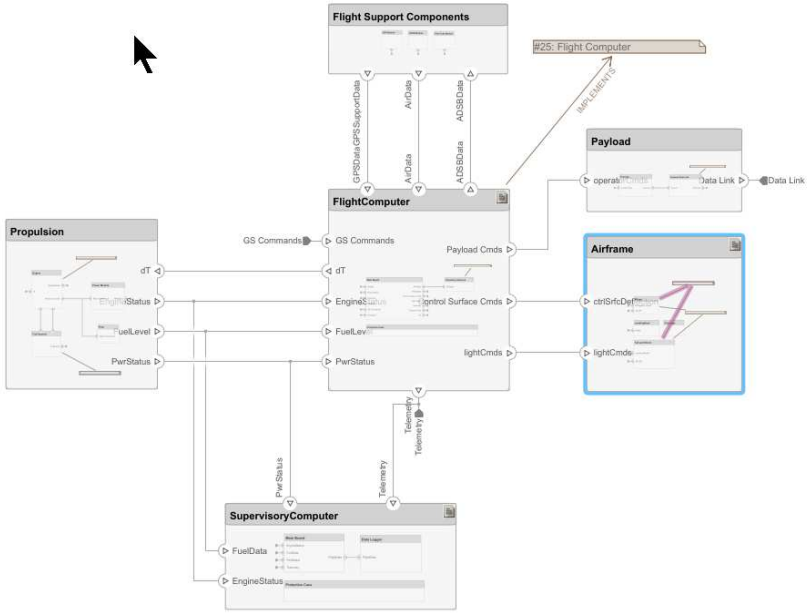


System Composer

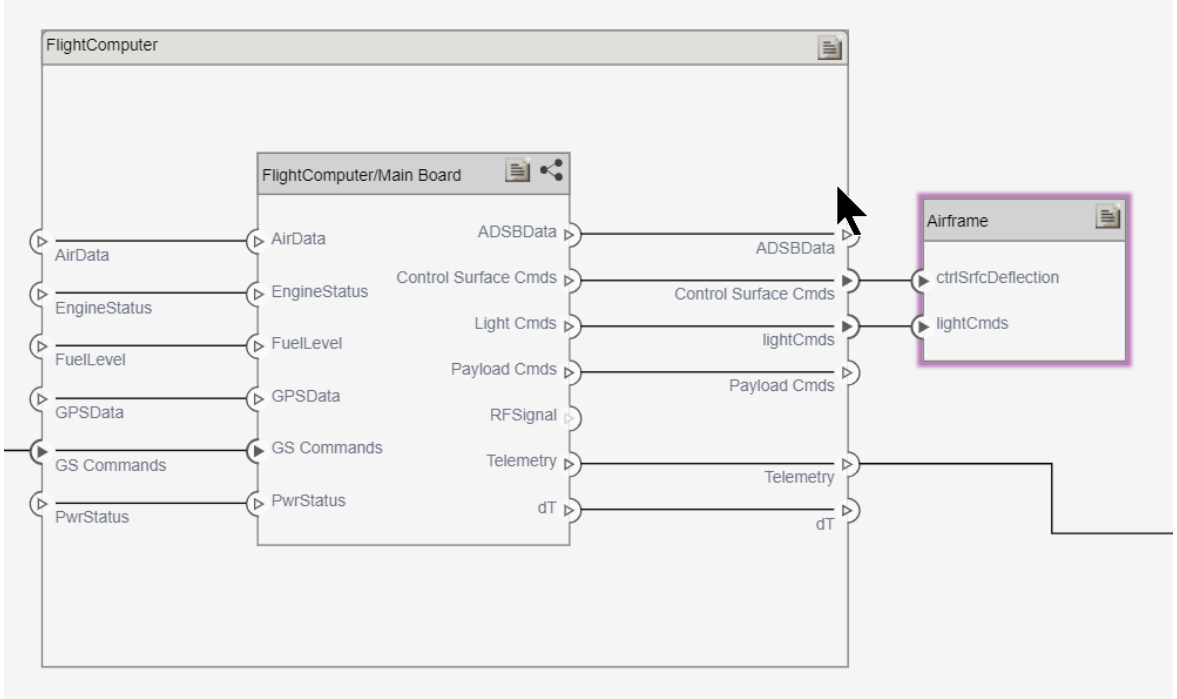
Tackle Architecture complexity with spotlight views

R2019a

Composition



Spotlight

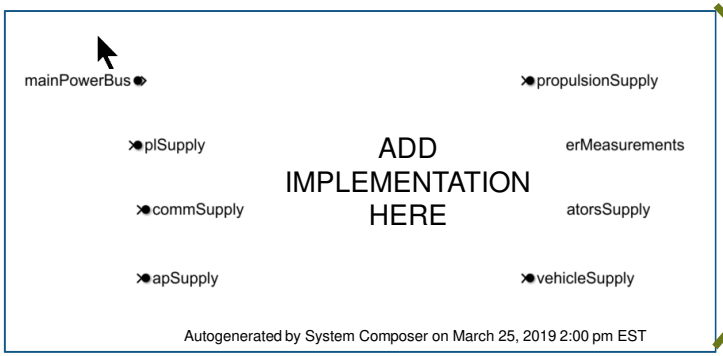


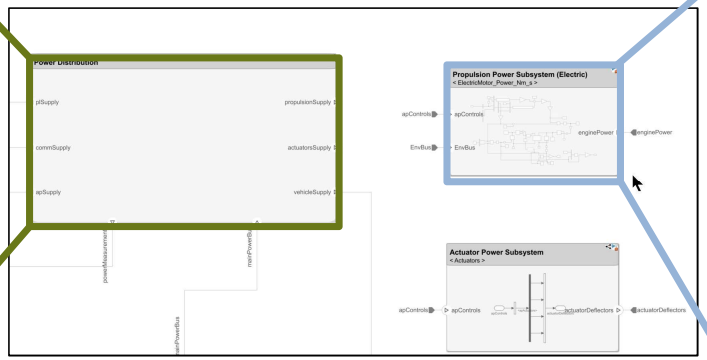
System Composer

R2019a

System and software architectures connected to implementations in Simulink

Generate Simulink models from architecture components





Link Simulink models to architecture components