

MATLAB EXPO 2019

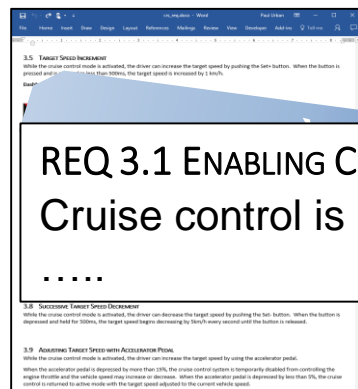
요구사항부터
아키텍처 설계와 시뮬레이션까지
시스템 엔지니어링을 위한 방안

류성연



Key Takeaways

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



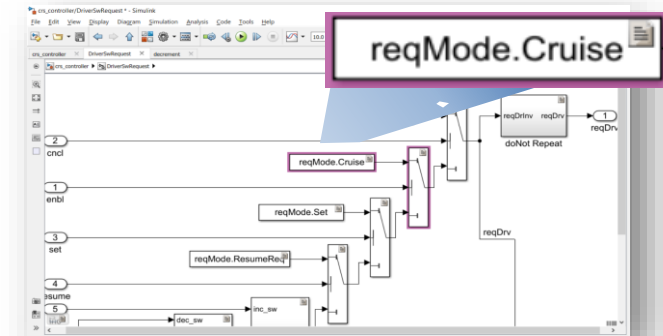
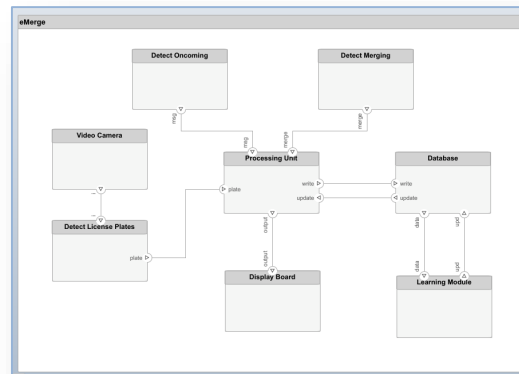
REQ 3.1 ENABLING CRUISE CONTROL
Cruise control is enabled when
.....

Derives

ENABLE SWITCH DETECTION
If the Enable switch is pressed
.....

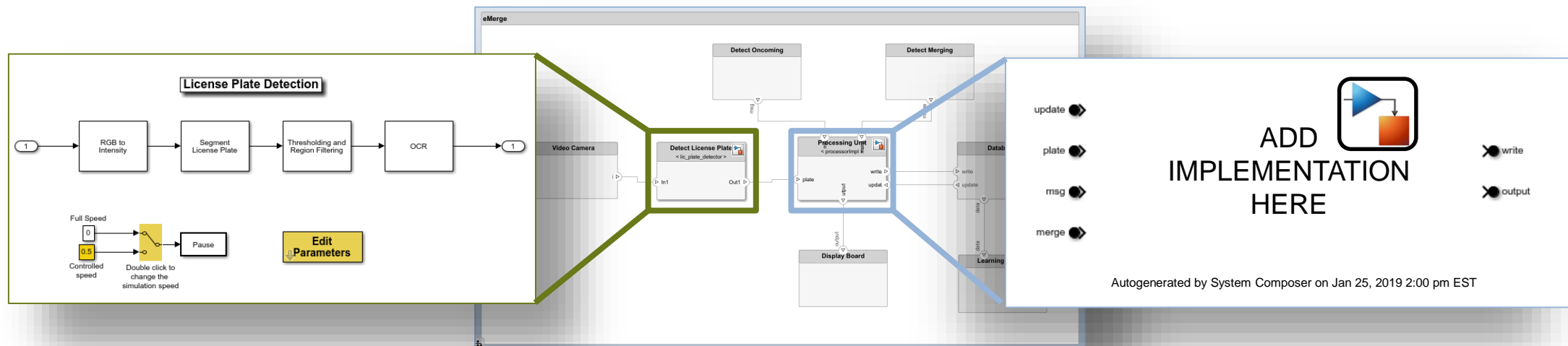
Implemented
By

Implemented
By



Key Takeaways

- Digital thread providing traceability between requirements, architecture, and design
- Connected environment for designing and analyzing architectures and designs



Key Takeaways

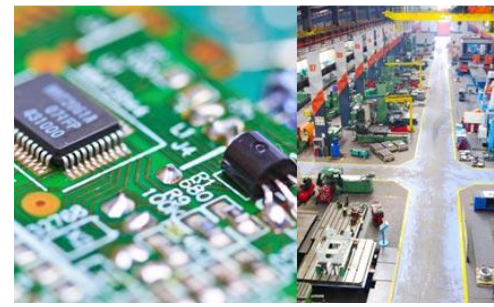
- Digital thread providing traceability between requirements, architecture, and design
- Connected environment for designing and analyzing architectures and designs
- Integrated platform for analyzing all parts of your architecture in one multi-domain environment



Dynamic Systems



State Machines



Discrete-Event



Physical Modeling

What does that mean?

Early in the Process Concepts/Descriptions

The image shows several hand-drawn diagrams and sticky notes. The top diagram is a block diagram of a system architecture with components: SERVER, APP, SUPERVISOR, SIGNAL PROCESSING, TRACKING, TARGET ACQ, DRONE CONTROL, and REFIDEL. Arrows indicate data flow between these components. Below this is a diagram of a drone system with blocks for Localization, Path Planning, Power System, and Drive System, connected by data lines like SelfLocation, MotorEncoderCount, DriveCom, and BatteryData. At the bottom left, there is a board with many colorful sticky notes. To the right, there is a text box containing technical specifications:

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)
The System shall determine the amount of residual oxygen present in the 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

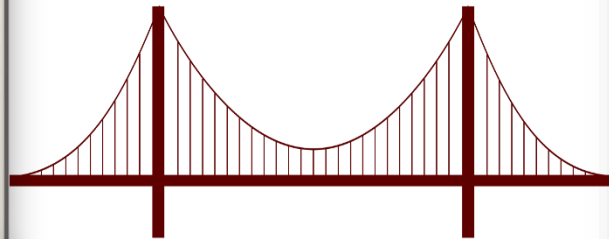
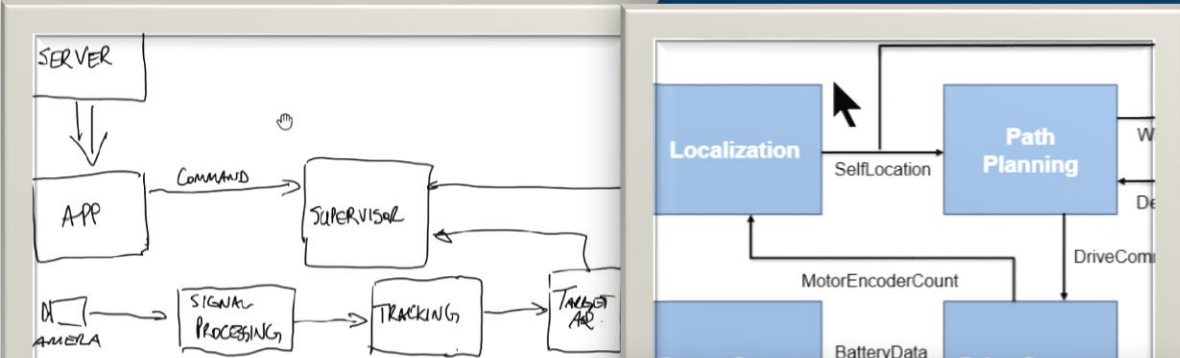
Later in the Process Models

The image shows a detailed block diagram representing a later-stage model. It features a central blue trapezoidal block connected to an orange square block. This central assembly is surrounded by numerous other blocks, some labeled 'wfp1Case_1000' and 'wfp2Case_1000', with a complex network of interconnecting lines representing data or signal flow.

What is the Gap?

Early in the Process
Concepts/Descriptions

Later in the Process
Models



Digital Thread
Connected Environment
Analysis & Simulation Platform

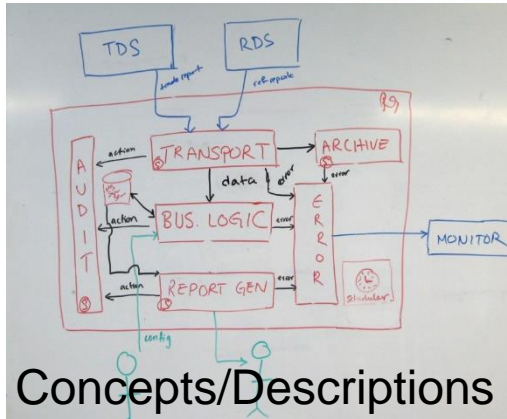
What goes into the bridge?

Be Intuitive

Facilitate Analysis

Tackle Complexity

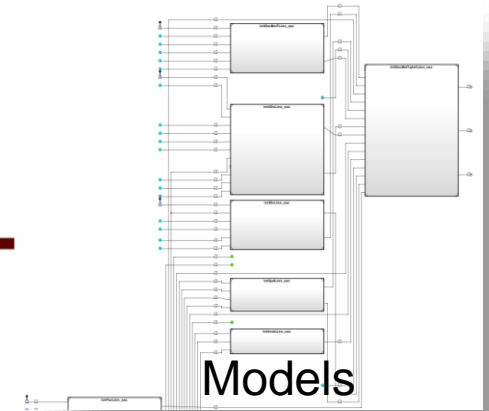
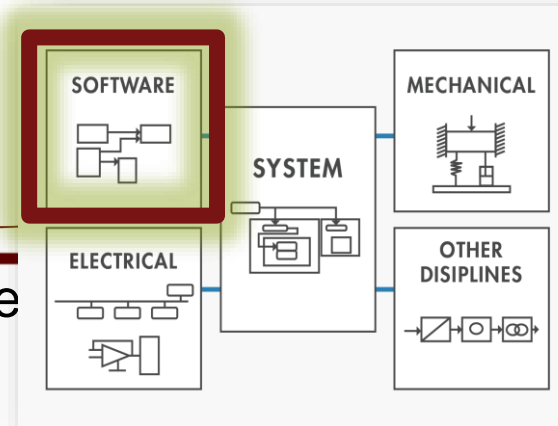
Enable Implementation



Concepts/Descriptions

VEHICLE COMPONENT	MASS(kg)	POWER(W)
• COMMUNICATION SUBSYS.	→ 2.63	58
- ADSB	→ 0.05	5
- KU/KA RADIO	→ 0.05	2
- RADIO RX PPM/PWM	→ 2.5	50
	→ 0.01	0.85
• ELECTRICAL SUBSYS	→ 0.02	1
- ACTUATOR POWER	533.15	353000
- POWER DISTRIBUTION	8	200
- POWER MONITORING	10	1000
- POWER SOURCE	→ 300	350000
- PROPULSION POWER	50	50
- VEHICLE POWER	5	0.02
- AUTOPILOT REGULATOR	0.05	1.07
- COMMS REGULATOR	0.05	1.07
• MONITORING + CONTROL SUBS.	3.55	1.150
- AUTOPILOT	0.6	1

anne



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)

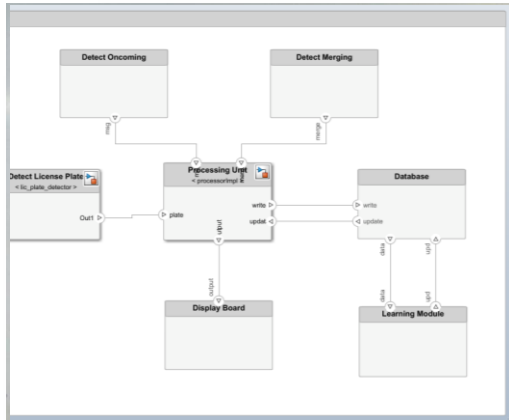
MathWorks Solution: System Composer R2019a and ...

✓ Be Intuitive

✓ Facilitate Analysis

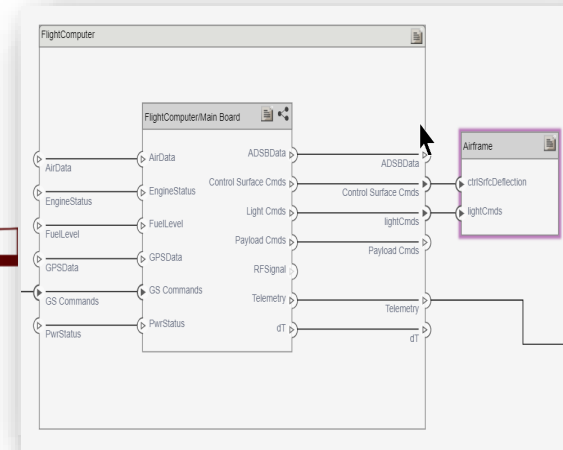
✓ Tackle Complexity

✓ Enable Implementation



VEHICLE COMPONENT

	MASS(kg)	POWER(W)
• COMMUNICATION SUBSYS.	→ 2.63	58
- ADSB	→ 0.05	5
- KU/Ka RADIO	→ 2.05	2
- RADIO RX PPM/PWM	→ 0.1	50
• ELECTRICAL SUBSYS	→ 0.02	0.85
- ACTUATOR POWER	→ 533.15	1
- POWER DISTRIBUTION	→ 8	353000
- POWER MONITORING	→ 10	200
- POWER SOURCE	→ 0.1	1000
- PROPULSION POWER	→ 300	350000
- VEHICLE POWER	→ 50	50
- AUTOPLOT REGULATOR	→ 5	0.02
- COMMS REGULATOR	→ 0.05	1.07
- MONITORING + CONTROL SUBS.	→ 0.05	1.07
- AUTOPLOT	→ 0.5	1.150
		1



Requirements Coverage Reporting and Impact Analysis

Simulink Requirements

Requirements - SmallUAV

View: Requirements

Index	Summary
> 1.1	Airworthiness
> 1.2	Communications
▼ 1.3	Payload Capabilities
1.3.1	Carrying Capacity
1.3.2	Payload Bay Capacity
1.3.3	Default Payload
1.3.4	Payload Protection

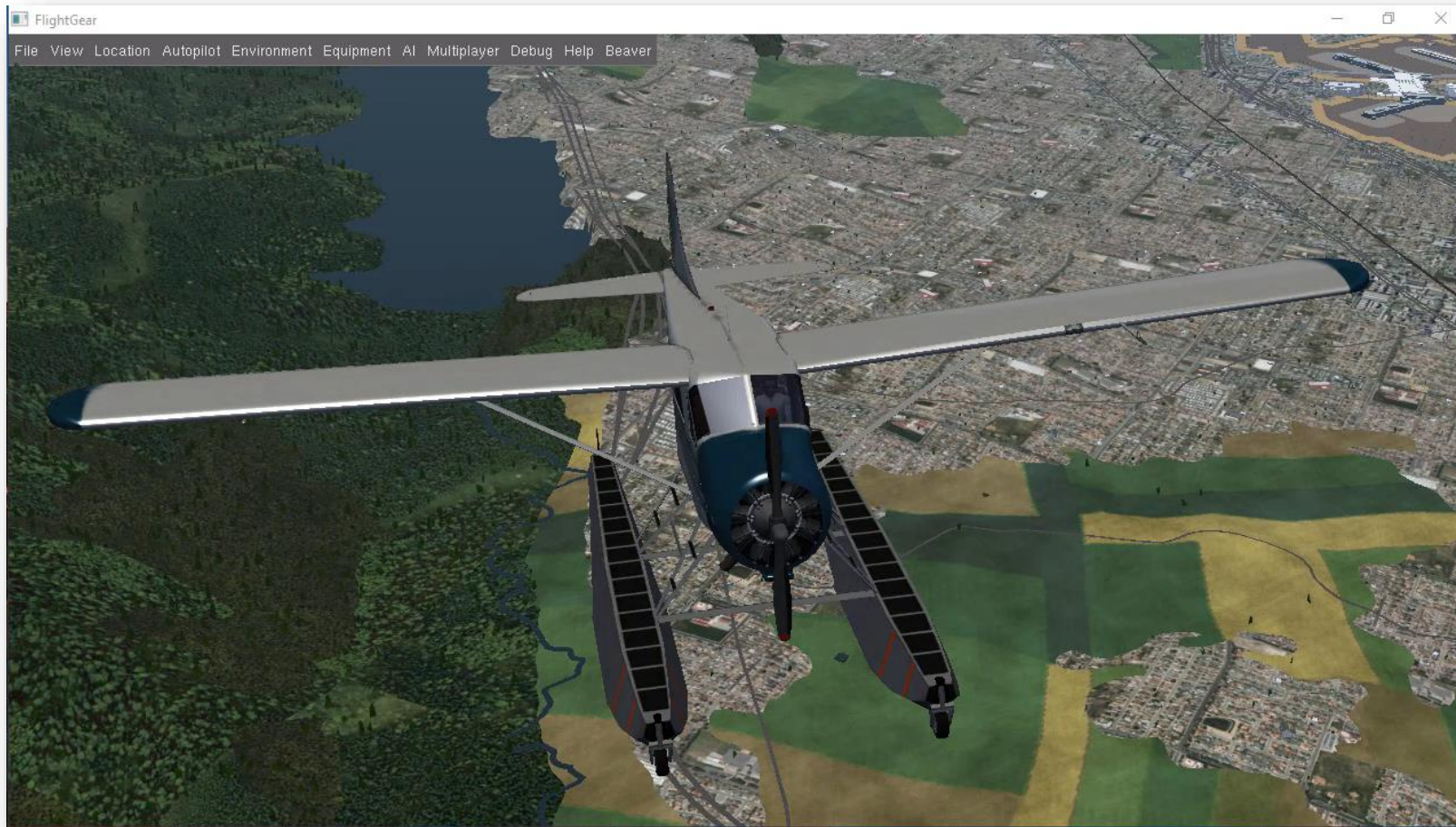
Implemented

Case Study: electrifying propulsion system

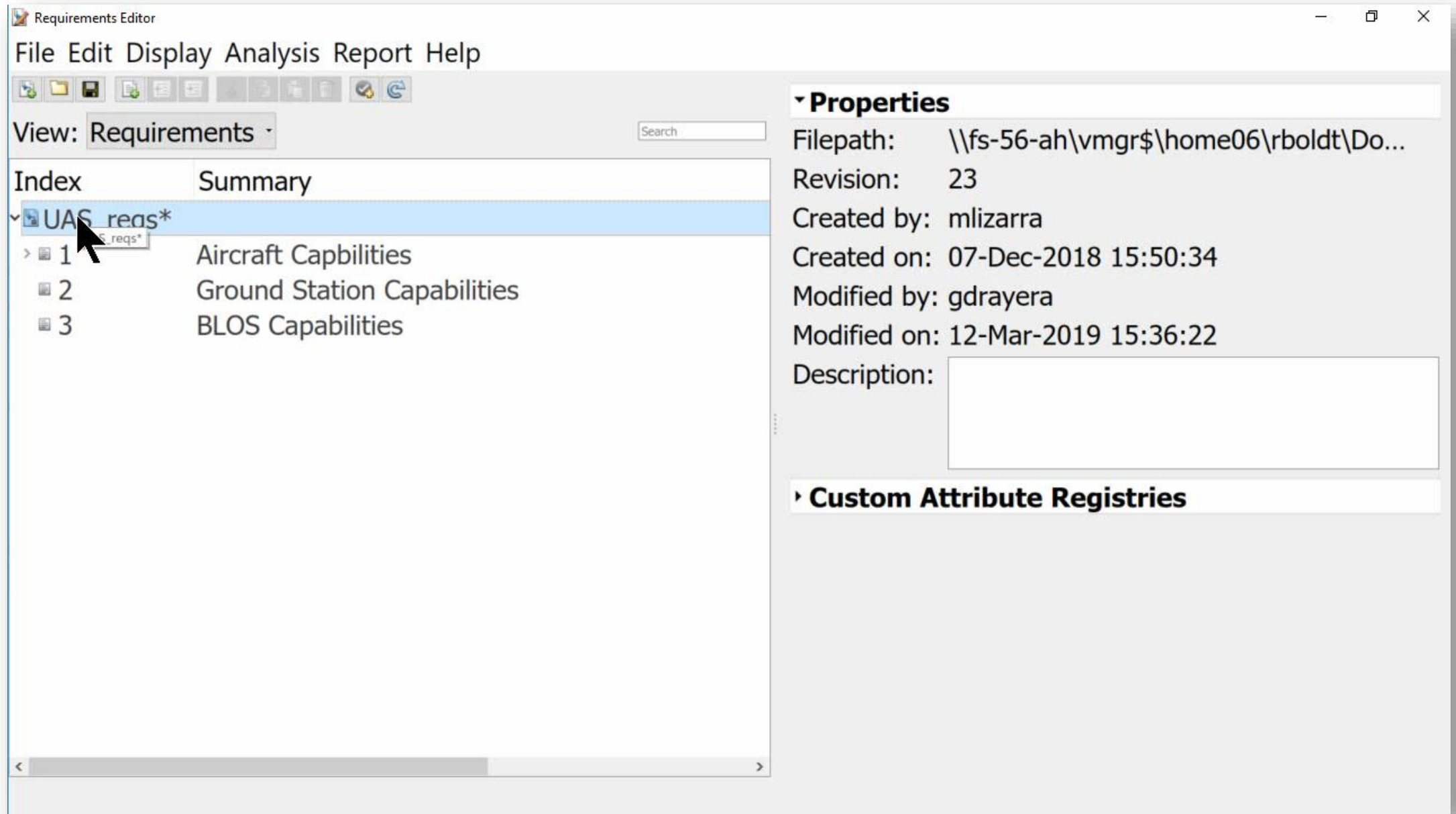
- System architecture using System Composer
- System requirement traceability
- Simulink modeling from system architecture
- System update for electrification
- Trade studies

Demo: De Havilland "Beaver" Airplane

- Target: Electrifying propulsion system



Start from Requirements



The screenshot shows the Requirements Editor application window. The title bar reads "Requirements Editor" and the menu bar includes "File", "Edit", "Display", "Analysis", "Report", and "Help". Below the menu bar is a toolbar with various icons. The main area is divided into two panes. The left pane, titled "View: Requirements", contains a tree view with a search box. The tree view shows a root node "UAS_reqs*" which is expanded to show three sub-nodes: "1 Aircraft Capabilities", "2 Ground Station Capabilities", and "3 BLOS Capabilities". A mouse cursor is pointing at the "1" icon next to the first sub-node. The right pane, titled "Properties", displays metadata for the selected item: Filepath: \\fs-56-ah\vmgr\$\home06\rboldt\Do..., Revision: 23, Created by: mlizarra, Created on: 07-Dec-2018 15:50:34, Modified by: gdrayera, Modified on: 12-Mar-2019 15:36:22, and a description field which is currently empty. Below the properties panel is a section titled "Custom Attribute Registries".

Index	Summary
UAS_reqs*	
1	Aircraft Capabilities
2	Ground Station Capabilities
3	BLOS Capabilities

Properties

- Filepath: \\fs-56-ah\vmgr\$\home06\rboldt\Do...
- Revision: 23
- Created by: mlizarra
- Created on: 07-Dec-2018 15:50:34
- Modified by: gdrayera
- Modified on: 12-Mar-2019 15:36:22
- Description:

Custom Attribute Registries

Easy to Design at a High Level of Abstraction

The screenshot shows the Simulink environment for a model named 'UAS_ref_arch'. The main workspace displays a high-level block diagram with three primary components: 'Ground Station', 'Vehicle', and 'BVLOS Navigation'. Each component is represented as a block with a header and a body. The 'Ground Station' block is connected to the 'Vehicle' block via a link labeled 'uaLOSdownlink'. The 'Vehicle' block is connected to the 'BVLOS Navigation' block via two links: 'uaBVLOSdownlink' and 'uaBVLOSuplink'. The 'BVLOS Navigation' block has two ports: 'uaBVLOSdownlink' and 'uaBVLOSuplink'. The 'Vehicle' block has two ports: 'uaBVLOSdownlink' and 'uaBVLOSuplink'. The 'Ground Station' block has a port labeled 'uaLOSdownlink'. The 'Model Browser' on the left side of the window is highlighted with a red box, showing various icons for navigating and editing the model. The status bar at the bottom indicates 'Ready', '105%' zoom, and 'VariableStepAuto'.

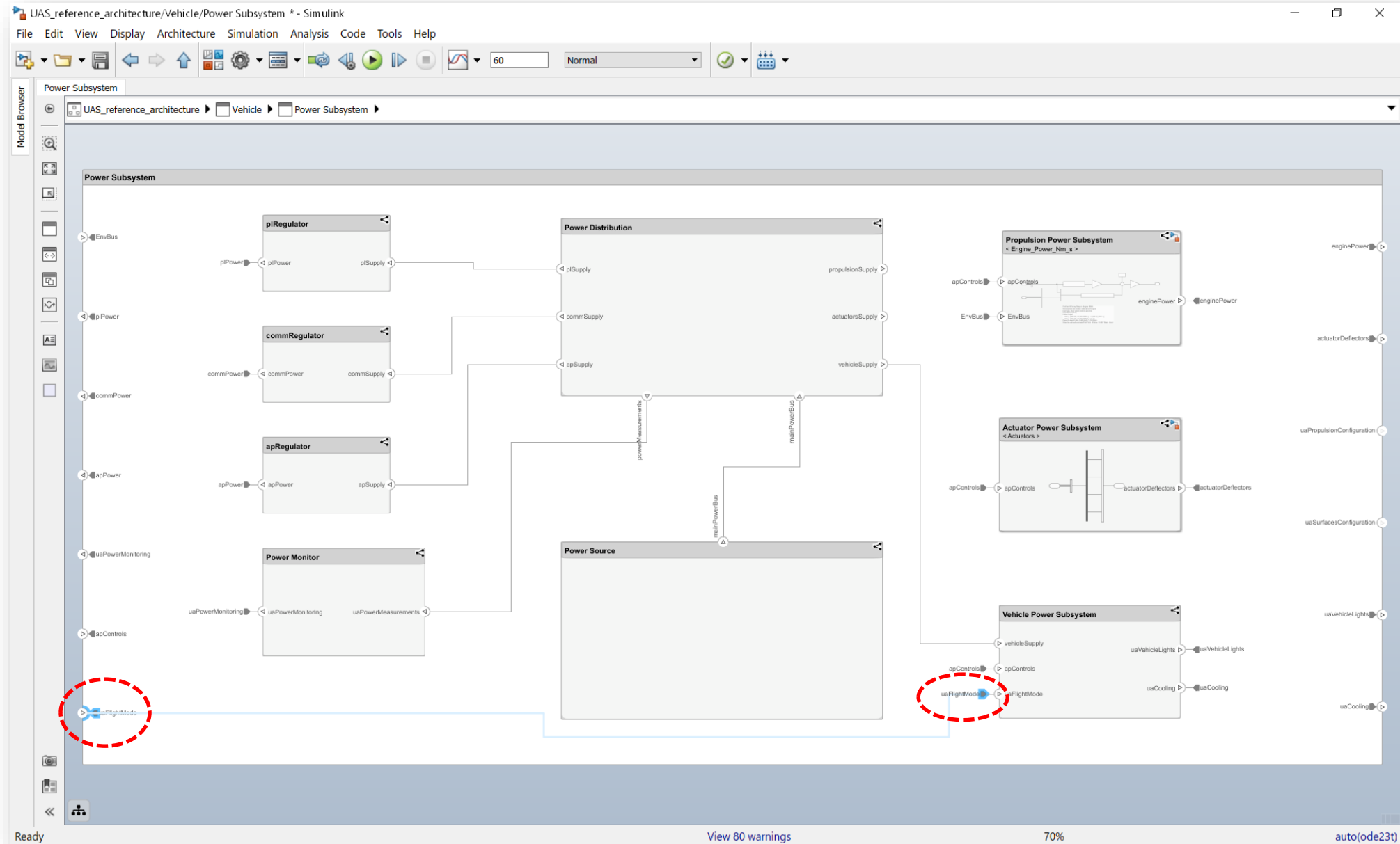
Add Details for Interfaces

The screenshot displays the Simulink environment for the 'UAS_ref_arch' model. The main workspace shows three interconnected blocks: 'Ground Station', 'Vehicle', and 'BVLOS Navigational'. Each block has a set of ports for communication. The 'Ground Station' block has ports for 'uaLOSdownlink', 'uaLOSuplink', 'gsBVLOSgownlink', and 'gsBVLOSuplink'. The 'Vehicle' block has ports for 'uaBVLOSdownlink', 'uaBVLOSuplink', 'uaLOSdownlink', and 'uaLOSuplink'. The 'BVLOS Navigational' block has ports for 'uaBVLOSdownlink', 'uaBVLOSuplink', and 'gsBVLOSgownlink'. Lines connect these ports to show the data flow between the components.

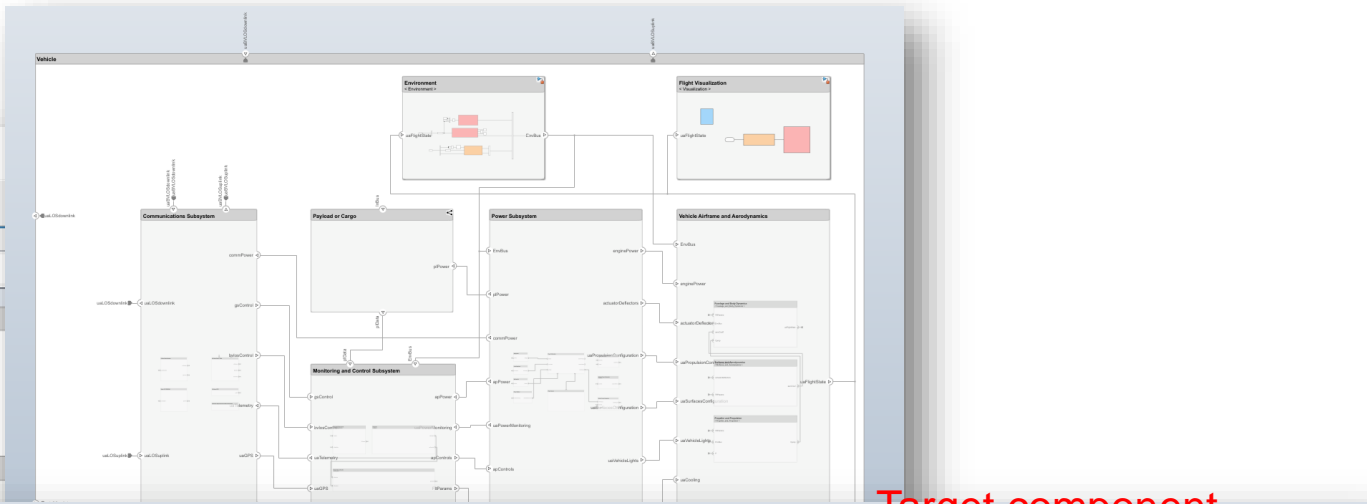
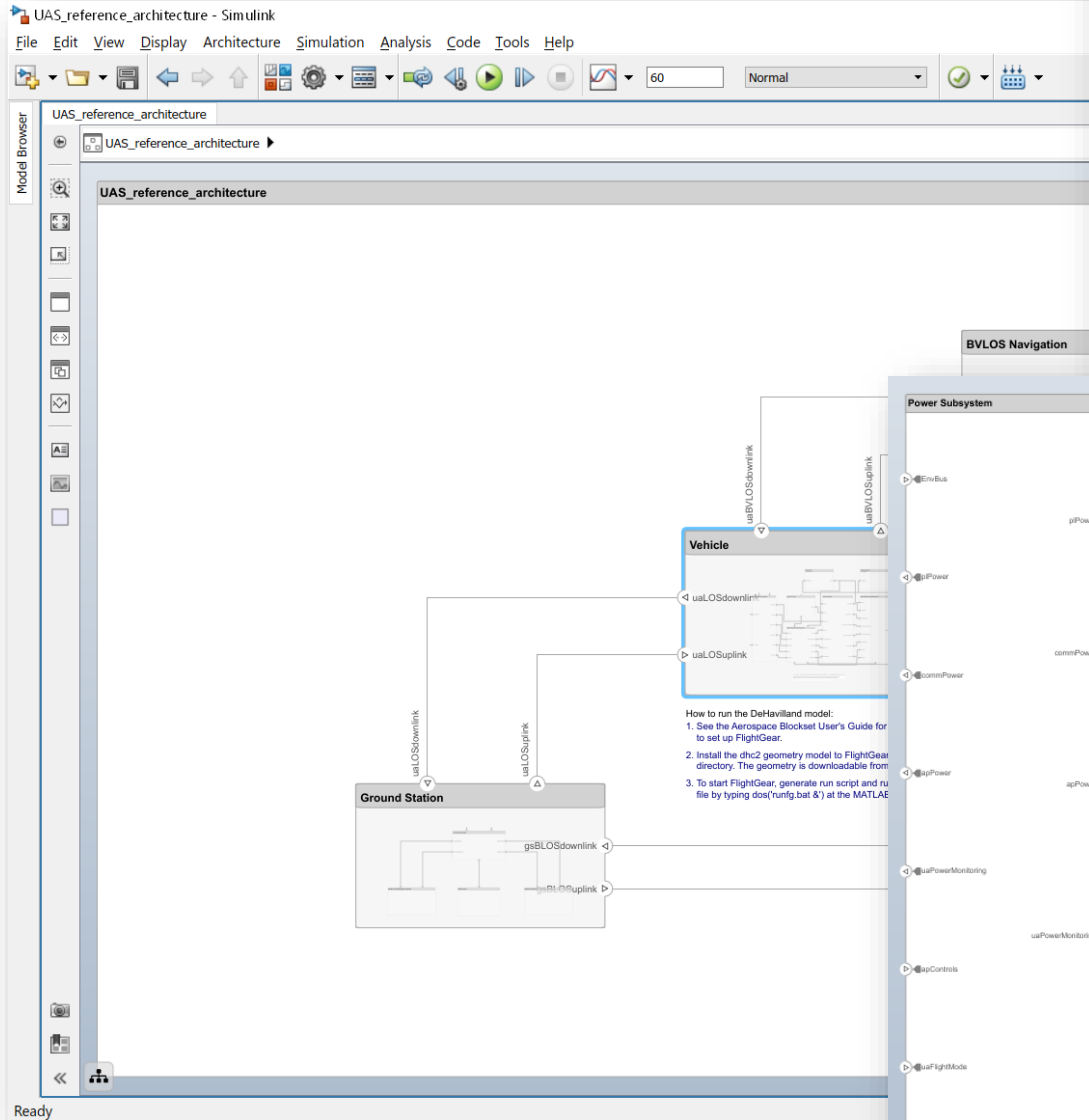
On the right side, the 'Interfaces' pane is open, showing the source 'UAS_ref_arch.slx' and a list of interfaces under the 'Telemetry' category. The 'Altitude' interface is selected. Below this, the 'Property Inspector' provides details for the 'Altitude' interface element:

Interface : Telemetry Element : Altitude	
Properties	
NAME	VALUE
Type	double
Dimensions	1
Units	feet
Complexity	real
Minimum	0
Maximum	30000
Description	Altitude of plane

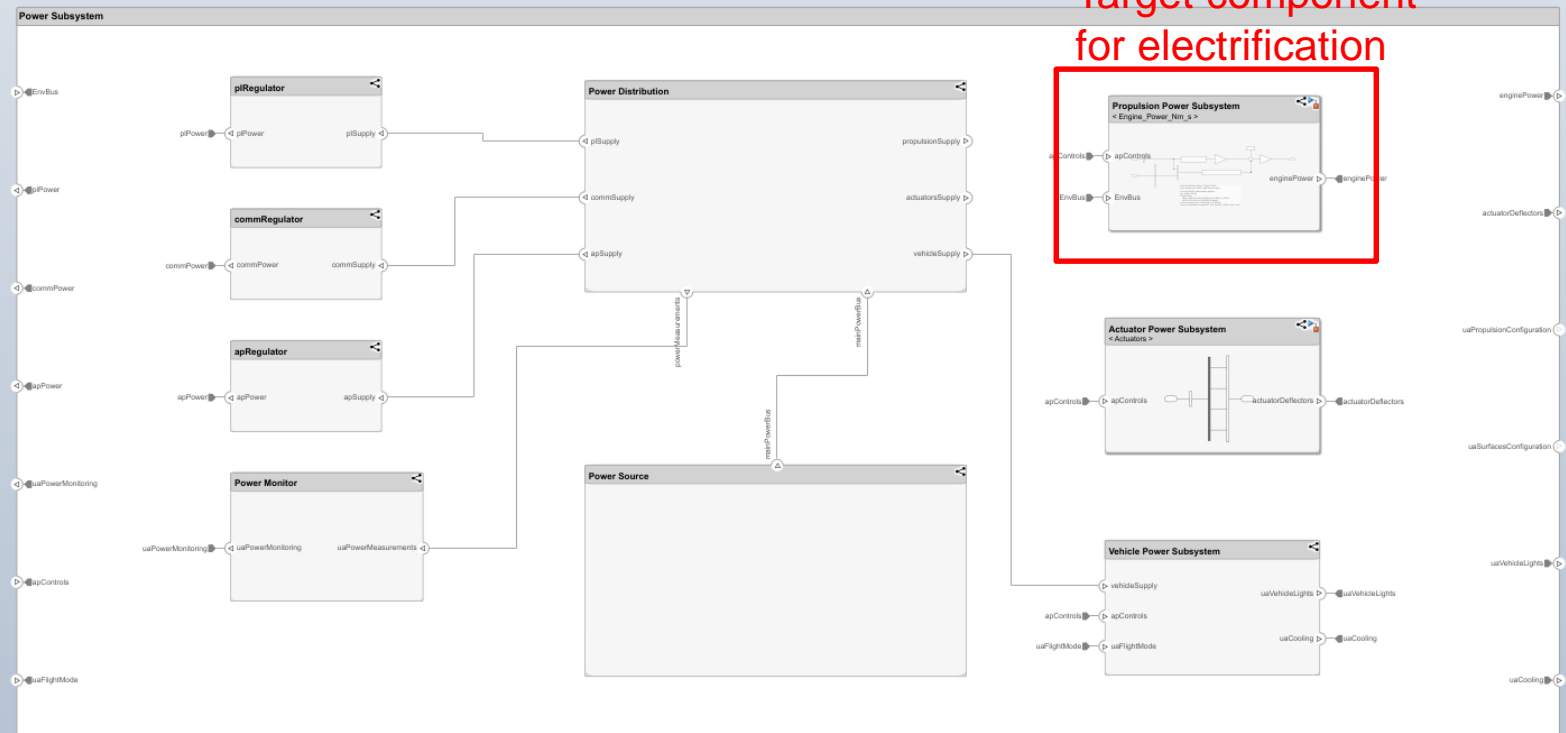
Automatic Simplified Signal Routing



Hierarchical System Design



Target component for electrification



Case Study

- System architecture using System Composer
- **System requirement traceability**
- Simulink modeling from system architecture
- System update for electrification
- Trade studies

Traceability with Simulink Requirements

The screenshot displays the Simulink Requirements tool interface. The top window shows a model diagram with a 'Vehicle' block and various requirement links. A blue arrow points from the 'Vehicle' block to the 'Construction' requirement in the table below. The bottom window shows a table of requirements with a red arrow pointing from the 'Propulsion Power' requirement to the 'Vehicle' block.

Index	Summary	Implemented
> 1.2	Communications	<div style="width: 100%; background-color: blue;"></div>
> 1.3	Payload Capabilities	<div style="width: 100%; background-color: blue;"></div>
▼ 1.4	Construction	<div style="width: 75%; background-color: blue;"></div>
1.4.1	Modularity	<div style="width: 100%; background-color: blue;"></div>
1.4.2	Propulsion Power	<div style="width: 0%; background-color: blue;"></div>
> 1.5	Flying Qualities	<div style="width: 100%; background-color: blue;"></div>
2	Ground Station Capabilities	<div style="width: 100%; background-color: blue;"></div>
3	BLOS Capabilities	<div style="width: 100%; background-color: blue;"></div>

Drilling Down to Propulsion Power Subsystem

UAS_reference_architecture - Simulink

File Edit View Display Architecture Simulation Analysis Code Tools Help

Model Browser

UAS_reference_architecture

Vehicle

uaLOdownlink

uaLOSuplink

How to run the DeHavilland model:

1. See the Aerospace Blockset User's Guide for instructions to set up FlightGear.
2. Install the dhc2 geometry model to FlightGear's data/Aircraft directory. The geometry is downloadable from www.flightgear.org.
3. To start FlightGear, open a console and run the generated batch...

Requirements - UAS_reference_architecture

View: Requirements

Index	Summary	Implemented
> 1.2	Communications	<div style="width: 100%; background-color: blue;"></div>
> 1.3	Payload Capabilities	<div style="width: 100%; background-color: blue;"></div>
▼ 1.4	Construction	<div style="width: 75%; background-color: blue;"></div>
1.4.1	Modularity	<div style="width: 100%; background-color: blue;"></div>
1.4.2	Propulsion Power	<div style="width: 0%; background-color: blue;"></div>
> 1.5	Flying Qualities	<div style="width: 100%; background-color: blue;"></div>

Property Inspector

Component

Architecture Info

NAME	VALUE
▼ Main	
Name	Vehicle
Stereotype	Add..

Interfaces

Ready 100% VariableStepAuto

Linking Requirement to Propulsion Power Subsystem

UAS_reference_architecture/Vehicle/Electrical Subsystem* - Simulink

File Edit View Display Architecture Simulation Analysis Code Tools Help

Model Browser

Electrical Subsystem

UAS_reference_architecture > Vehicle > Electrical Subsystem

#35: Propulsion Power

IMPLEMENTS

Propulsion Power Subsystem
< Engine_Power_Nm_s >

apControls

EnvBus

enginePower

propulsionSupply

actuatorsSupply

vehicleSupply

Property Inspector

Component

Architecture Info

NAME	VALUE
Main	
Name	Electrical Subsystem
Stereotype	Add..

Requirements - UAS_reference_architecture

View: Requirements

Index	Summary	Implemented
UAS_reqs		
1	Aircraft Capabilities	
1.1	Airworthiness	
1.1.1	Range	
1.1.2	Rain Conditions	
1.1.3	Power	

Ready 125% VariableStepAuto

Case Study

- System architecture using System Composer
- System requirement traceability
- **Simulink modeling from system architecture**
- System update for electrification
- Trade studies

Create Simulink from System Composer

The screenshot illustrates the process of creating a Simulink model from a System Composer architecture. The main window shows the 'UAS_reference_architecture/Vehicle/Power Subsystem' model. A context menu is open over the 'enginePower' component, with 'Create Simulink Behavior...' highlighted in red. A dialog box titled 'Create Simulink behavior' is displayed, prompting the user to specify a behavior model name. The 'New model name' field contains 'Engine_Power_Nm_s'. The 'OK' button is highlighted with a blue box. A blue arrow points from the 'OK' button to the 'PropulsionPowerSubsystem' window.

The 'PropulsionPowerSubsystem' window shows the resulting Simulink model. The 'Model Browser' on the left lists the components: 'PropulsionPowerSubsystem', 'apControls', 'EnvBus', and 'enginePower'. The 'Property Inspector' on the right shows the component's properties:

NAME	VALUE
Main	
Name	Propulsion Power Subsystem
Stereotype	Add..

The Simulink diagram shows the 'enginePower' component connected to 'apControls' and 'EnvBus'.

Link Simulink Model to System Composer

The screenshot shows the MATLAB/Simulink environment with the following elements:

- Model Browser:** Displays the project hierarchy. The 'Link to Model...' option is highlighted with a red box.
- Link to model dialog:** A modal dialog box with the title 'Link to model'. It contains the text 'Link to the specified model.' and a text field for 'Model name:' containing 'Engine_Power_Nm_s'. There are 'Browse...', 'OK', 'Cancel', and 'Help' buttons.
- Engine_Power_Nm_s Simulink Model:** A Simulink block diagram for a Pratt and Whitney Wasp Jr. Engine R-985. It features two input buses: 'apControls' (port 1) and 'EnvBus' (port 2). The diagram includes several mathematical blocks:
 - A gain block with value 0.00412 .
 - A summing junction with value 326.5 .
 - A gain block with value 0.7355 .
 - A gain block with value 1 .
- Property Inspector:** Shows the 'Propulsion Power Subsystem' component with a 'Name' property set to 'Propulsion Power Subsystem'.
- Text Description:**

Pratt and Whitney Wasp Jr. Engine R-985
 Nine-cylinder, air-cooled, radial aircraft engine
 Fuel type: 80/87 grade aviation gasoline
 Dry weight: 290 kg
 Power output:
 400 hp (298 kW) at 2,200 RPM up to 5,000 ft (1,500 m);
 450 hp (336 kW) at 2,300 RPM for takeoff
 Power-to-weight ratio: 0.625 hp/lb (1.03 kW/kg)
https://en.wikipedia.org/wiki/Pratt_%26_Whitney_R-985_Wasp_Junior

Simulink Model Traceability

The screenshot shows the Simulink environment for a model named 'UAS_reference_architecture/Vehicle/Electrical Subsystem'. The main workspace contains a 'Propulsion Power Subsystem' block, which is a subsystem of type '< Engine_Power_Nm_s >'. This block has two input ports: 'apControls' and 'EnvBus', and one output port: 'enginePower'. A red box highlights the top-right corner of the 'Propulsion Power Subsystem' block, where a callout box labeled '#35: Propulsion Power' is connected by an arrow labeled 'IMPLEMENTS'. The 'Property Inspector' on the right side of the window shows the following details for the selected component:

Component	
Architecture Info	
NAME	VALUE
Main	
Name	Propulsion Power Subsystem
Stereotype	Add..
SubsystemBudget	Select

The status bar at the bottom of the window shows 'Ready', '125%', and 'VariableStepAuto'.

Define Profiles and Stereotypes for Trade Studies

- Define non-functional properties on elements in an architecture model to verify structural and functional requirements

Profile: System Standard

Stereotype: System General

Property: Element ID
Property: Cost

Stereotype: System Component

Property: Development cost
Property: Required hardware
Property: Development Time

Stereotype: Physical Connector

Property: Length
Property: Unit cost
Property: Material

The screenshot shows the Simulink System Composer Profile Editor interface. The 'Architecture' menu is open, highlighting the 'Profile' option. The 'System Composer Profile Editor' window is active, displaying the configuration for a stereotype named 'SubsystemBudget'.

System Composer Profile Editor
Describe architecture profiles, stereotypes and custom property sets for use with System Composer architecture models. [show more...](#)

Profile: New Profile Open Save Close Stereotype: New Stereotype Close Import into model: Select model ?

Profile Browser: Filter profiles by model: <all>

- UAVComponent
 - SubsystemBudget

Stereotype Properties:

- Name: SubsystemBudget
- Applies to: Component
- Base stereotype: <nothing>
- Abstract stereotype
- Description: Represents the base component of UAVComponent

Property name	Type	Name	Unit	Default
1 Mass	double	n/a	kg	0
2 Power	double	n/a	mW	0

Define Profiles and Stereotypes

The screenshot shows the Simulink environment with the following components:

- Model Browser:** Shows the hierarchy: UAS_reference_architecture > Vehicle > Electrical Subsystem.
- Diagram:** A central block titled "Propulsion Power Subsystem" with the stereotype "< Engine_Power_Nm_s >". It has three input ports: "apControls", "EnvBus", and "enginePower". An arrow labeled "IMPLEMENTS" points from the block to a note "#35: Propulsion Power". Below it is another block titled "Actuator Power Subsystem" with the stereotype "< Actuators >".
- Property Inspector:** A window on the right showing the properties of the selected "Propulsion Power Subsystem" component.

Component	
Architecture Info	
NAME	VALUE
Main	
Name	Propulsion Power Subsystem
Stereotype	Add..
SubsystemBudget	
Mass	290 kg
Power	350000000 mW

Simulation in System Composer

The screenshot displays the Simulink System Composer environment for a vehicle simulation. The main workspace shows a hierarchical block diagram of the vehicle system, including sub-systems like Environment, Flight Visualization, Communications Subsystem, Payload or Cargo, Electrical Subsystem, and Vehicle Airframe and Aerodynamics. A FlightGear window is overlaid on the model, showing a 3D rendering of a large aircraft flying over a landscape. The FlightGear window has a menu bar with options: File, View, Location, Autopilot, Environment, Equipment, AI, Multiplayer, Debug, Help, and Beaver. The status bar at the bottom indicates the simulation is running, with a progress bar showing 26% completion, a time of T=15.600, and a zoom level of 50%. The text 'auto(ode23t)' is visible in the bottom right corner.

Case Study

- System architecture using System Composer
- System requirement traceability
- Simulink modeling from system architecture
- **System update for electrification**
- Trade studies

Propulsion System Change to Electrified System

UAS_reference_architecture/Vehicle/Electrical Subsystem * - Simulink

File Edit View Display Architecture Simulation Analysis Code Tools Help

Electrical Subsystem

Model Browser

UAS_reference_architecture > Vehicle > Electrical Subsystem >

#35: Propulsion Power
IMPLEMENTS

Propulsion Power Subsystem
< Engine_Power_Nm_s >

apControls → apControls

EnvBus → EnvBus

enginePower → enginePower

Property Inspector

Requirement: #35

Details

▼ Properties

Type: Functional

Index: 1.4.2

Custom ID: #35

Summary: Propulsion Power

Description Rationale

The original gas engine of the aircraft shall be replaced by an equivalent output electrical motor, able to supply at least 350 kW of mechanical power at 2,300 RPM.

Keywords:

Revision information:

▼ Links

Implemented by:

Propulsion Power Subsystem

▼ Comments

Requirements - UAS_reference_architecture

View: Requirements

Index	Summary	Implemented
1.4	Construction	
1.4.1	Modularity	
1.4.2	Propulsion Power	
1.5	Flying Qualities	
2	Ground Station Capabilities	

Ready 125% VariableStepAuto

Spotlight Views

The screenshot displays the Simulink environment for a UAS reference architecture. The main workspace shows a block diagram of the 'Electrical Subsystem' containing a 'Propulsion Power Subsystem' (labeled '< Engine_Power_Nm_s >') and an 'Actuator Power Subsystem' (labeled '< Actuators >'). A context menu is open over the Propulsion Power Subsystem, with the 'Create Spotlight From Component' option highlighted. The menu includes options like 'Explore', 'Open As Top Model', 'Cut', 'Copy', 'Paste', 'Delete', 'Add Variant Choice', 'Apply Stereotype', 'Format', 'Arrange', 'Signals & Ports', 'Requirements', 'Block Parameters (ModelReference)', 'Properties...', and 'Help'. The right side of the interface shows a 'Property Inspector' and 'Interfaces' panel with signals like 'enginePower' and 'actuatorDeflectors'.

- Needs for analyze large and complex system
 - Generating simplified spotlight with filtering
 - Capturing upstream and downstream dependencies

Spotlight View Change to Another

UAS_reference_architecture/Vehicle/Electrical Subsystem * - Simulink

File Edit View Display Architecture Simulation Analysis Code Tools Help

Electrical Subsystem

UAS_reference_architecture

Vehicle/Electrical Subsystem

Environment

uaFlightState EnvBus

Vehicle Bus CAN 2.0b

apActuators apControls

apPropulsion

Propulsion Power Subsystem

EnvBus enginePower

apControls

uaFlightMode

Propeller and Propellers

EnvBus FitParams Cprop

P

OutBus

actuatorDeflectors

apPower

commPower

plPower

uaCooling

uaPowerMonitoring

uaPropulsionConfiguration

uaSurfacesConfiguration

uaVehicleLights

enginePower

Spotlight

Model Browser

Interfaces

Property Inspector

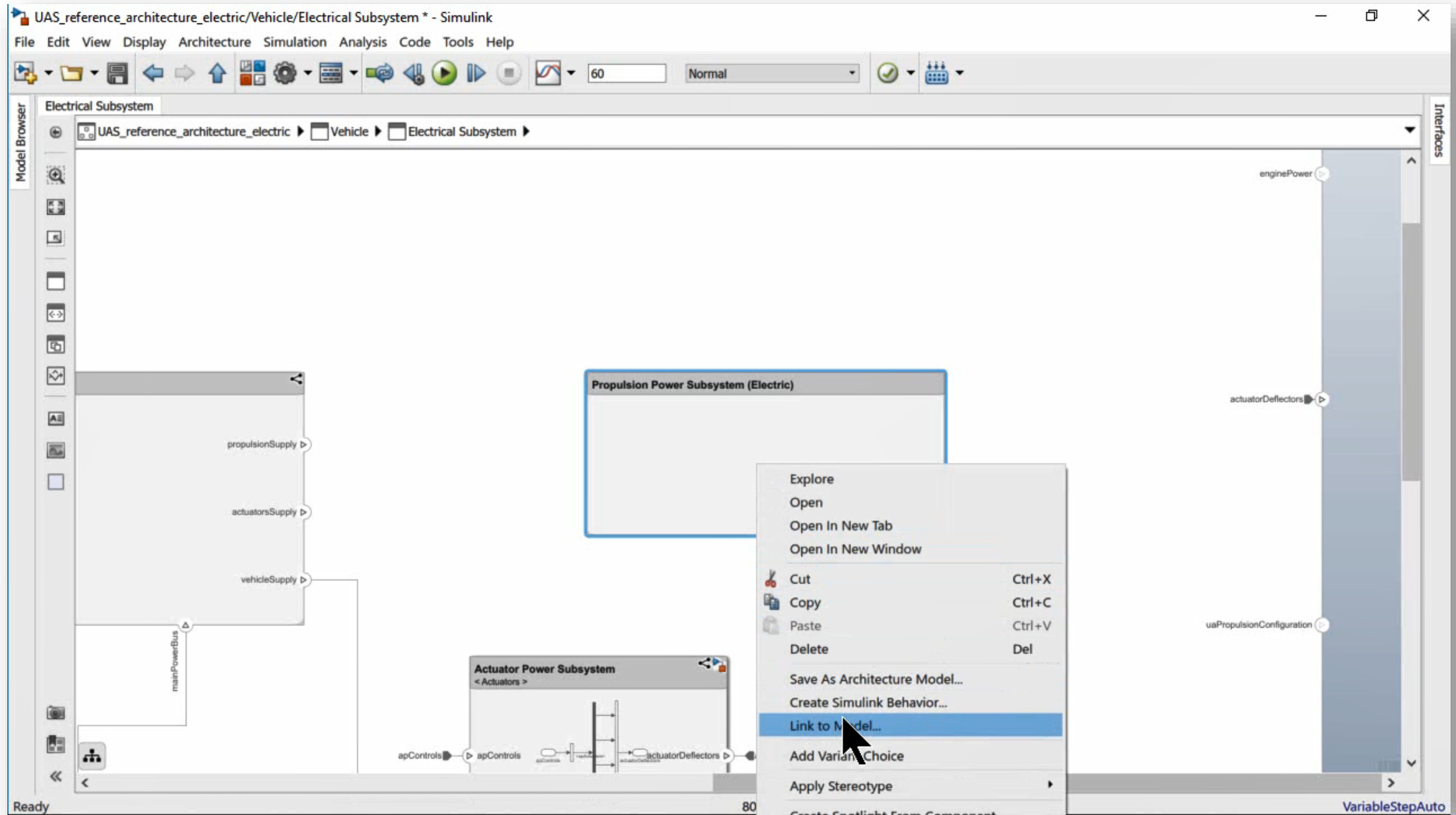
Ready

80%

VariableStepAuto

-(black): Components connected in same level
 -(grey): Components connected in different level

Replace Simulink Models in System Composer



Case Study

- System architecture using System Composer
- System requirement traceability
- Simulink modeling from system architecture
- System update for electrification
- **Trade studies**

Early architectural decisions often have non-functional implications

+ Electric motor

+ Battery



What is the impact of extra weight on the range of the flight time?

Non-Functional Properties



Stereotype Change for Impact Analysis

The screenshot shows the Simulink environment with a model titled "UAS_reference_architecture_electric/Vehicle/Electrical Subsystem". The main workspace displays a block diagram of the "Propulsion Power Subsystem (Electric)" which is a subsystem of type "ElectricMotor_Power_Nm_s". This subsystem has two input ports: "apControls" and "EnvBus", and one output port: "enginePower". Below it, the "Actuator Power Subsystem" is partially visible, with an output port "uaPropulsionConfigur".

The "Property Inspector" on the right side of the window shows the configuration for the selected "Propulsion Power Subsystem (Electric)" component. It is currently in the "Info" tab. The properties are organized into sections:

- Main**
 - Name: Propulsion Power S...
 - Stereotype: Add..
- SubsystemBudget**
 - Mass: 100 kg
 - Power: 175000000 mW

The "Power" property is currently selected and highlighted in blue.

Trade Studies

The screenshot displays the Simulink environment for a trade study. The 'Architecture' menu is open, showing options for creating analysis models. The 'Property Inspector' on the right provides a table of component properties.

NAME	VALUE
Main	
Name	Propulsion Power S...
Stereotype	Add..
SubsystemBudget	
Mass	100 kg
Power	175000000 mW

Trade Studies

Analysis Viewer (Technical Preview)

HOME

New Open Save Delete Analyze Arguments BottomUp Update Automatic Overwrite

INSTANCE MODEL ANALYSIS UPDATE

Instances	Mass	Power
UAS_reference_architecture_electric_budgetRollup	392.33	200614300
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	200614300
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	200355090
Actuator Power Subsystem	8	300000
Power Distribution	10	1000
Power Monitor	0	0
Power Source	20	1000
Propulsion Power Subsystem (Electric)	100	200000000
Vehicle Power Subsystem	5	50000
apRegulator	0.05	20
commRegulator	0.05	1070
piRegulator	0.05	2000
Environment	0	0

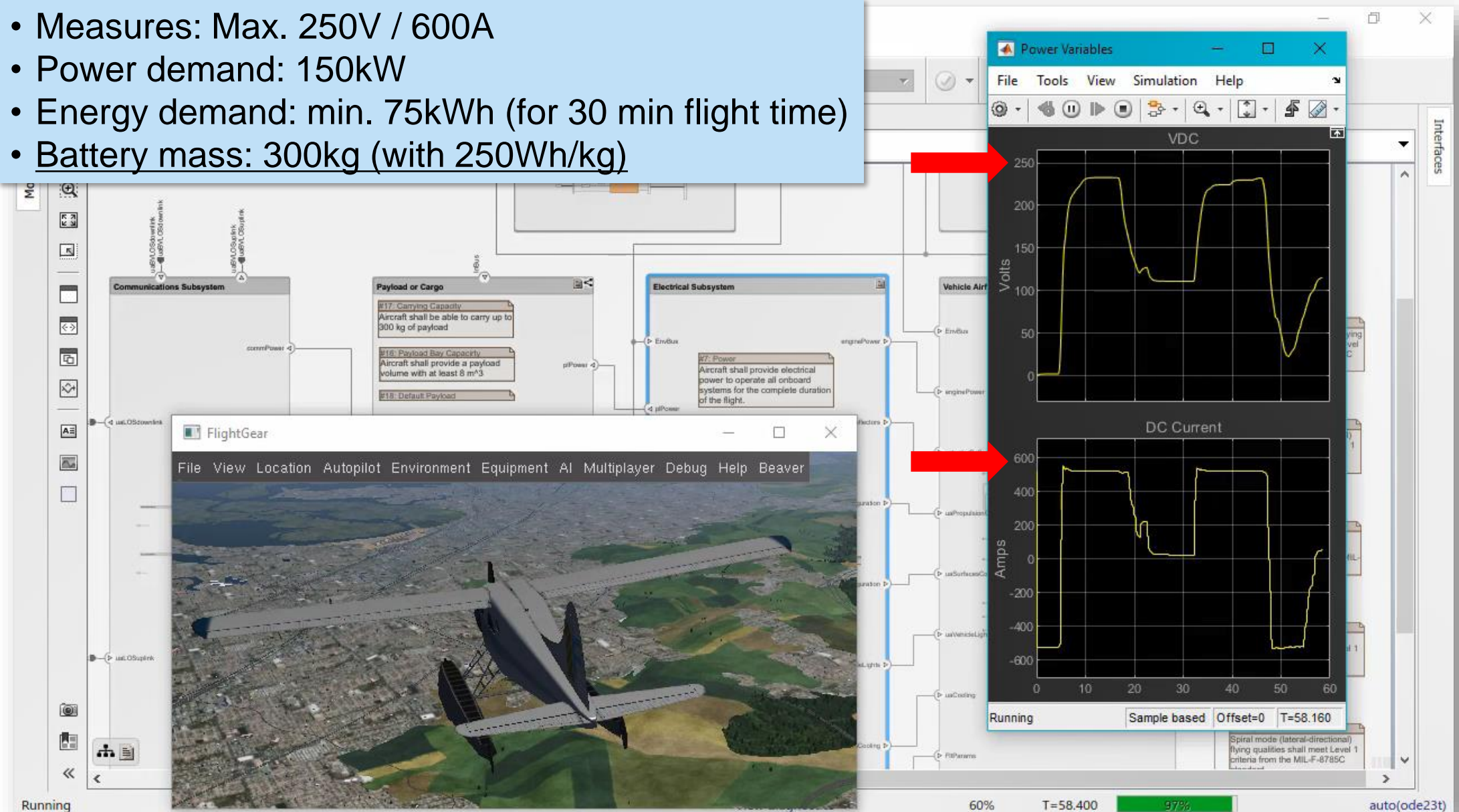
INSTANCE PROPERTIES

NodeInstance: Propulsion Power Subsystem (Electric)

Property	Units	Value	Edit
SubsystemBudget			
Mass	kg	100	
Power	mW	200,000,000	

Simulation with Electrified Propulsion Power System

- Measures: Max. 250V / 600A
- Power demand: 150kW
- Energy demand: min. 75kWh (for 30 min flight time)
- Battery mass: 300kg (with 250Wh/kg)





System Composer

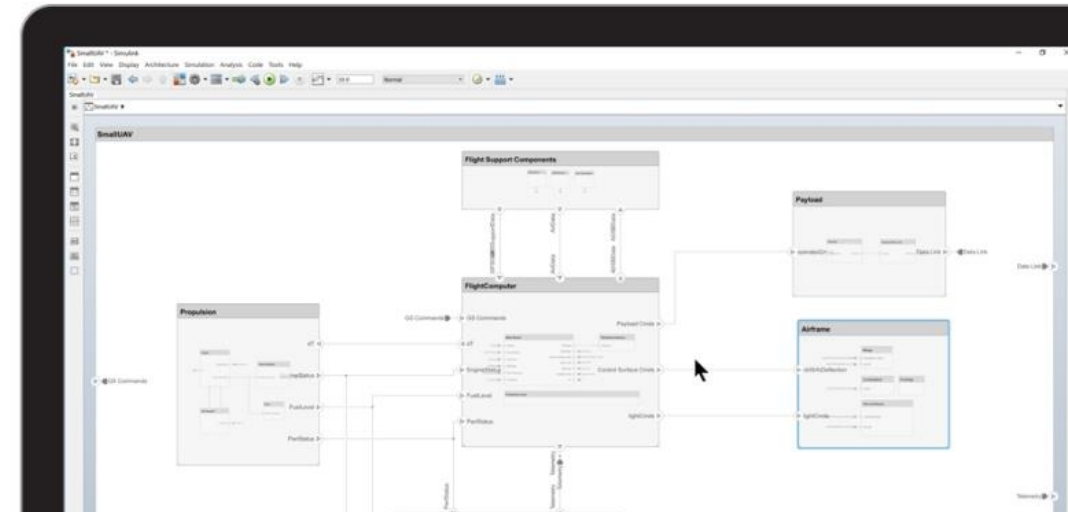
Design and analyze system and software architectures

[Download a free trial](#)

<https://kr.mathworks.com/products/system-composer.html>

System Composer™ enables the definition, analysis, and specification of architectures and compositions for model-based systems engineering and software design. With System Composer, you allocate requirements while refining an architecture model that can then be designed and simulated in Simulink®.

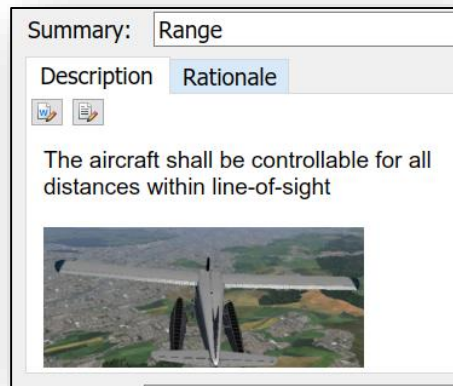
System Composer lets you create or import architecture models that describe a system in terms of components and interfaces. You can also populate an architecture model from the architectural elements of Simulink designs or C/C++ code. You can create custom live views of the model to study specific design or analysis concerns. With these architecture models you can analyze requirements, capture properties via stereotyping, perform trade studies, and produce specifications and ICDS.



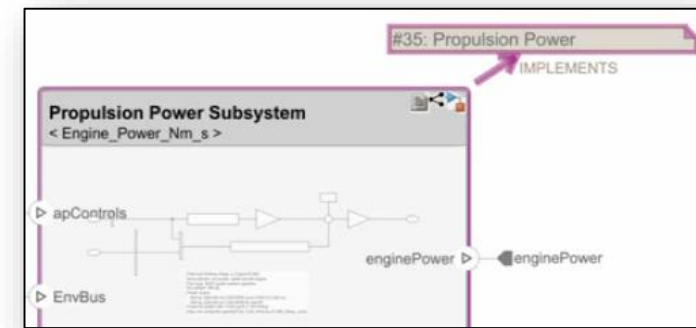
Digital Thread from Requirements to Architecture and Design

Simulink Requirements

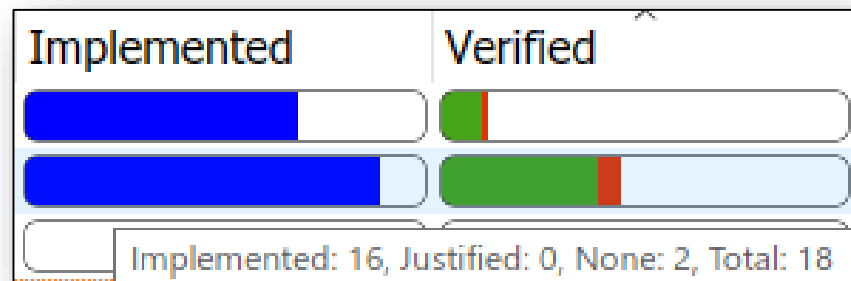
Author requirements or view from external source



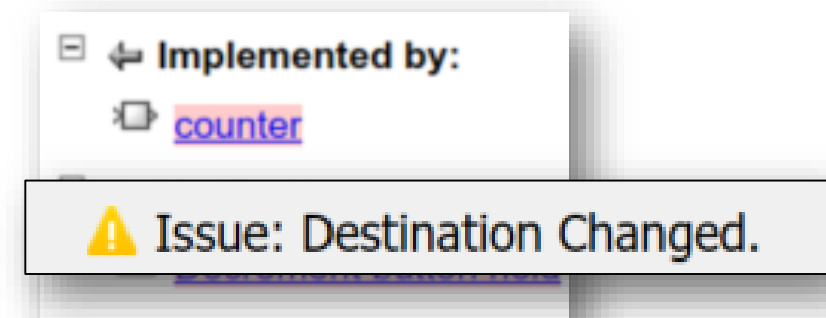
Link requirements, architectures, design, code and test



Identify gaps in architecture or design

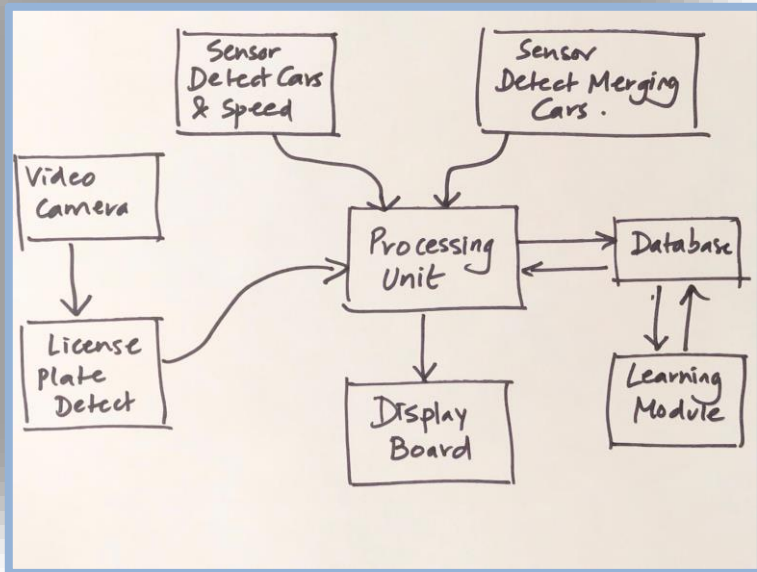


Identify impact of requirement changes

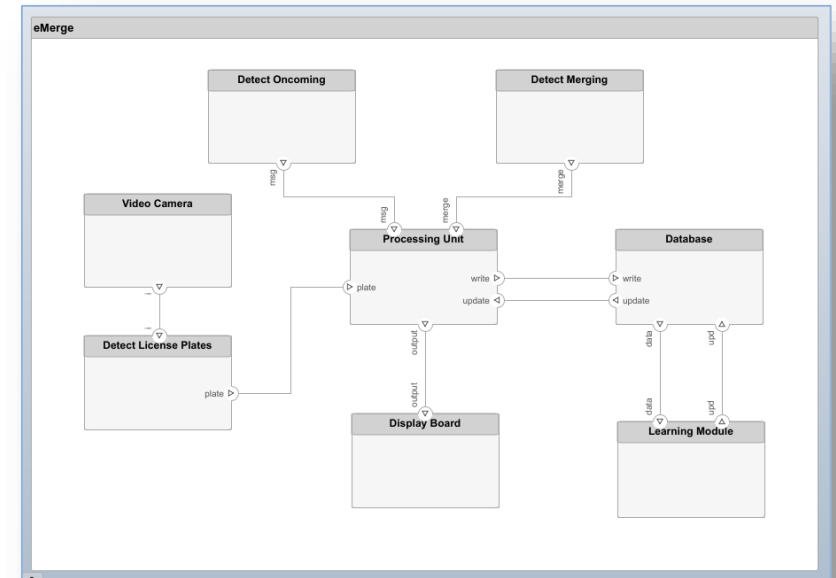


Intuitively design system and software architectures

System Composer



Description
=
Architecture



Perform trade studies based on data driven analysis to optimize architectures

Add custom data

Architecture	Info
NAME	VALUE
▼ Main	
Name	Power Unit
Stereotype	Add..
▼ OnboardElement Select	
Mass	0.217 kg
Power	0 mW
RFHarnessLength	0 cm

Create analysis model

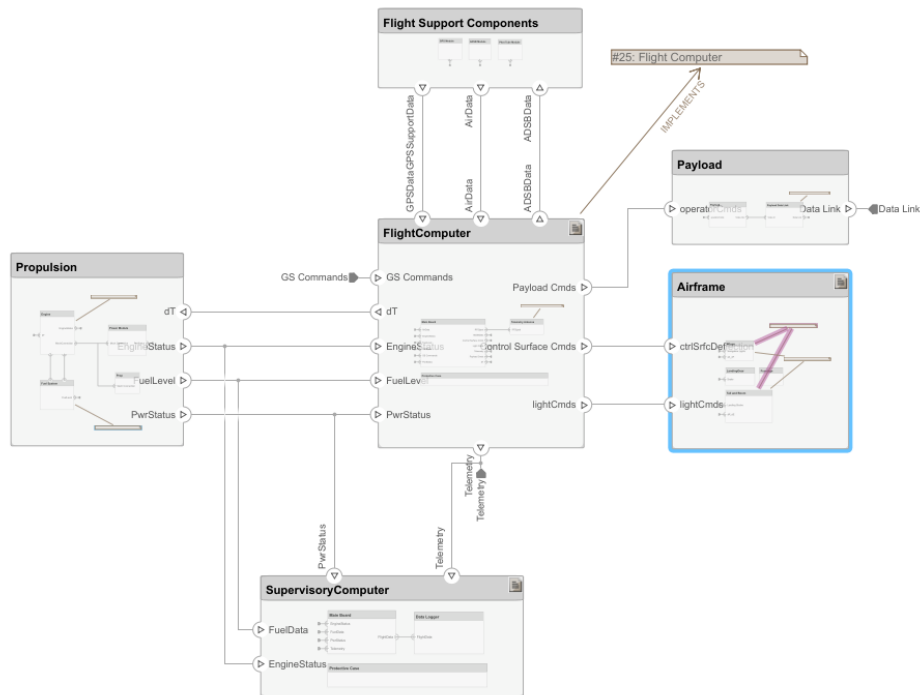
Instances	Mass(kg)
SmallUAV	0
Airframe	0
Fuselage	1.7
LandingGear	1.65
Tail and Boom	2.7
Wings	3.2
Flight Support Components	0
ADSB Module	0
ABDSB Antenna	0.058
ADSB Board	0.098
GPS Module	0
GPS Antenna	0.128
GPS Board	0.27
Pitot Tube Module	0.075
FlightComputer	0
Main Board	0.145
Protective Case	0.195

Calculate mass roll-up data

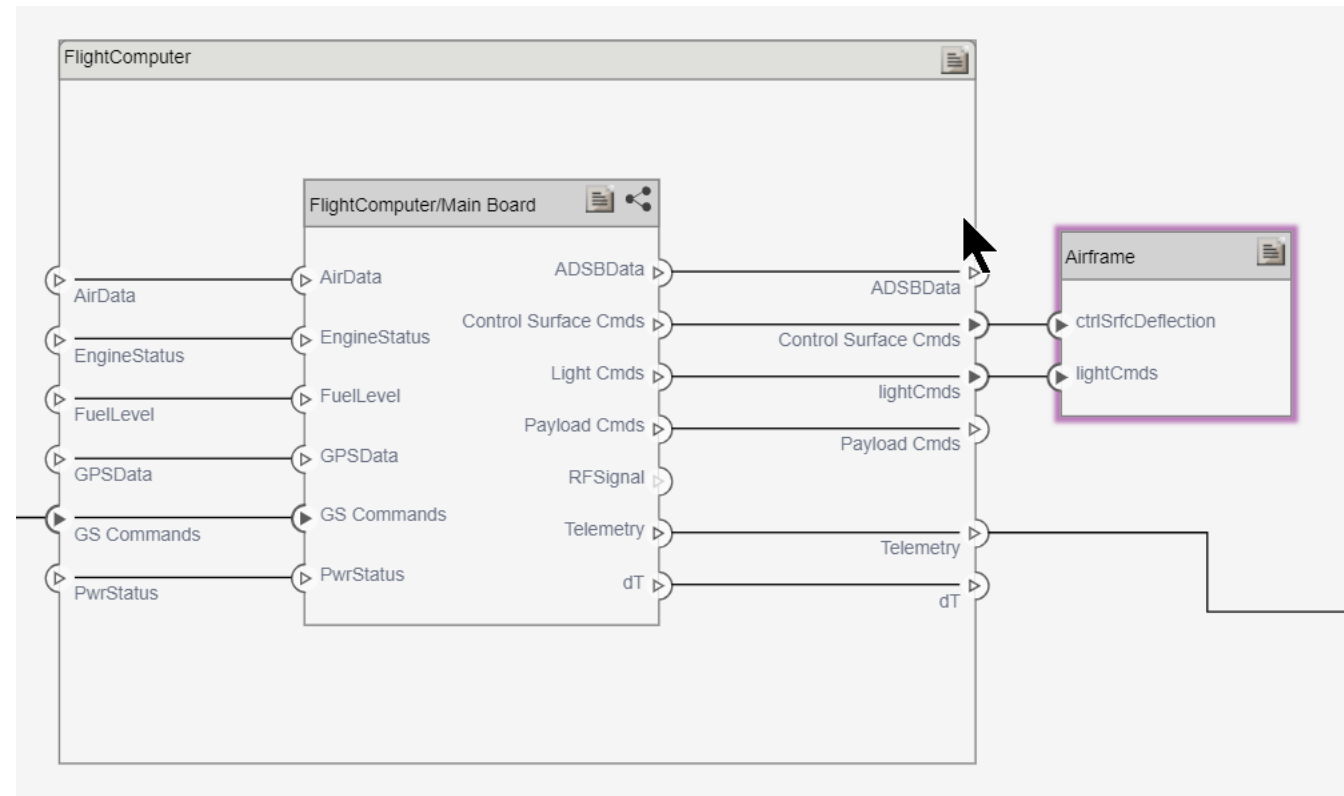
Instances	Mass(kg)
SmallUAV	15.932
Airframe	9.25
Fuselage	1.7
LandingGear	1.65
Tail and Boom	2.7
Wings	3.2
Flight Support Components	0.629
ADSB Module	0.156
ABDSB Antenna	0.058
ADSB Board	0.098
GPS Module	0.398
GPS Antenna	0.128
GPS Board	0.27
Pitot Tube Module	0.075
FlightComputer	0.388
Main Board	0.145
Protective Case	0.195

Tackle Architecture complexity with spotlight views

Composition



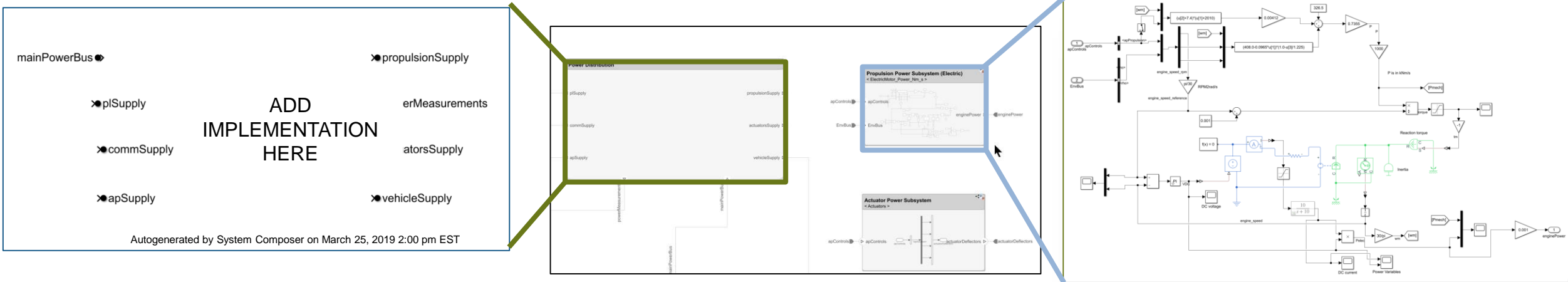
Spotlight



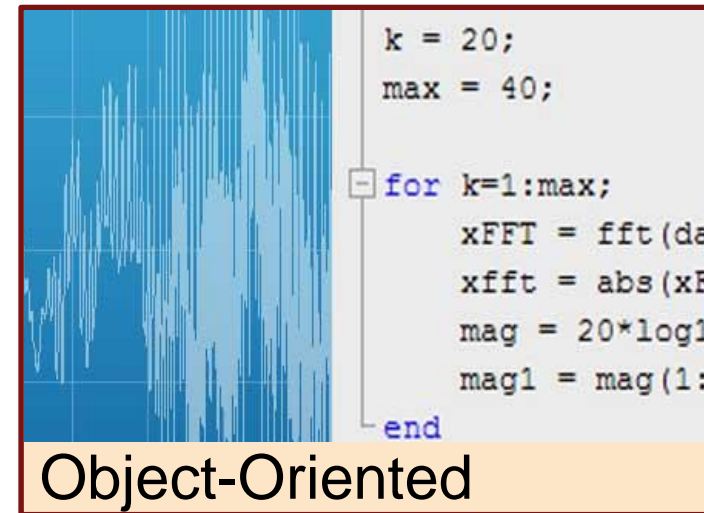
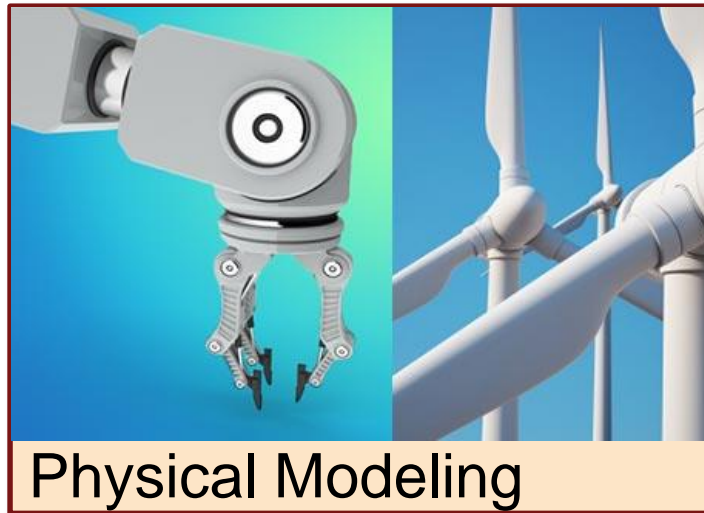
System and software architectures connected to implementations in Simulink

Generate Simulink models from architecture components

Link Simulink models to architecture components



Simulink: A Multi-Language Simulation Environment



Learn More

- [Simulink Requirement Webpage](#)
- [System Composer Webpage](#)
- [System Modeling and Simulation Webpage](#)

MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다

