Connecting System Architecture to Model-Based Design

Lyle Shipton
Application Engineer
MathWorks
Plano, TX
Agenda

- MathWorks Overview
- System Architecting and Model-Based Design
  - User Needs
  - Current workflows
  - MathWorks solution
  - Example
- Summary
- Resources
Background

- University of Illinois at Urbana-Champaign
  - B.S, M.S. Aerospace Engineering
- SpaceX Rocket Development Facility
  - Test Engineer
  - Lead Engineer, Integration & Test
- Eaton Aerospace, Fuel and Motion Controls
  - Lead Aerospace Systems Engineer
- MathWorks, Application Engineering Group
  - Lead Engineer, Aerospace applications
MathWorks at a Glance

- Privately held
- 4500 employees worldwide
- More than 4 million users in 185 countries

- Office locations
- Distributors serving 16 countries

Earth's topography on a Miller cylindrical projection, created with MATLAB and Mapping Toolbox.
Key Industries

- Aerospace and defense
- Automotive
- Biological sciences
- Biotech and pharmaceutical
- Communications
- Electronics
- Energy production
- Financial services
- Industrial automation and machinery
- Medical devices
- Metals, materials, and mining
- Neuroscience
- Railway systems
- Semiconductors
- Software and internet
Core MathWorks Products

**MATLAB**


- Designed for engineers and scientists
- Professionally developed, tested, and documented
- Toolboxes for:
  - Machine learning, data analytics, deep learning, image processing and computer vision, signal processing and communications, computational finance, robotics and control systems
- Interactive apps that automatically generate programs
- Easily scales to clusters, GPUs, and clouds
- Direct deployment to production enterprise applications
- Automatic conversion to embeddable C and CUDA code
- Integrates with Simulink to support Model-Based Design
Core MathWorks Products

Simulation and Model-Based Design

Model and simulate your system
- Use one multi-domain environment
- Model the system under test and the plant
- Simulate how all parts of the system behave

Test early and often
- Test your system under all conditions
- Validate your design with real-time testing
- Trace from requirements to design to code

Automatically generate code
- Generate production-quality C and HDL code
- Deploy directly to embedded processors or FPGA’s/ASIC’s
**Key capabilities for engineers and scientists**

<table>
<thead>
<tr>
<th>Technical Computing</th>
<th>Verification, Validation, and Test</th>
<th>Automatic Code Generation</th>
<th>System Modeling and Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB</td>
<td>MATLAB</td>
<td>Simulink</td>
<td></td>
</tr>
<tr>
<td>Application deployment</td>
<td>Test and measurement</td>
<td>Embedded code</td>
<td>Rapid prototyping and HIL</td>
</tr>
<tr>
<td>Student version</td>
<td>Model checking</td>
<td>Hardware support packages</td>
<td>DSP designs Communications systems</td>
</tr>
<tr>
<td>Instrument and database connectivity</td>
<td>Code verification</td>
<td>PLC code HDL code</td>
<td>State charts</td>
</tr>
<tr>
<td>Parallel computing</td>
<td>Code verification</td>
<td>MATLAB to C/HDL</td>
<td>Physical modeling</td>
</tr>
<tr>
<td>MATLAB Mobile for phones/tablets</td>
<td>Test automation</td>
<td>HW/SW co-development</td>
<td>Discrete-event simulation Video processing</td>
</tr>
<tr>
<td>MATLAB Online</td>
<td>Requirements authoring &amp; mgmt.</td>
<td>GPU code</td>
<td>Computer vision</td>
</tr>
<tr>
<td>Robotics and autonomous systems</td>
<td>MATLAB</td>
<td>RF Phased array</td>
<td>Machine learning</td>
</tr>
<tr>
<td>WLAN/LTE protocols</td>
<td>HDL verification</td>
<td>Big data</td>
<td>Deep learning</td>
</tr>
<tr>
<td>MATLAB Online</td>
<td>HW/SW co-development</td>
<td>AWS &amp; Azure support</td>
<td>Sensor fusion</td>
</tr>
<tr>
<td>HDL verification</td>
<td>ThingSpeak for IoT</td>
<td>Enterprise integration</td>
<td>Text analytics</td>
</tr>
</tbody>
</table>

**Matlab Works founded in 1984**
MathWorks Product Overview

Event-Based Modeling
Real-Time Simulation and Testing

Physical Modeling
Verification, Validation, and Test
Simulation Graphics and Reporting

SIMULINK®
Simulation and Model-Based Design

Applications
- Control Systems
- Signal Processing and Communications
- Image Processing and Computer Vision
- Test and Measurement
- Computational Finance
- Computational Biology

MATLAB®
The Language of Technical Computing

Parallel Computing
Code Generation

Math, Statistics, and Optimization
Application Deployment
Database Access and Reporting
How is this done today?

System of Systems

Textual requirements
How is this done today?

System of Systems

Architecture sketches are difficult/impossible to trace to requirements
How is this done today?

System of Systems

Architecture models traced to the requirements
How is this done today?

System of Systems

Trade studies determine acceptable/optimal architectures
How is this done today?

System of Systems

SoS Analysis

Design models traced to requirements
How is this done today?

System of Systems

SoS Analysis

Design models traced to architecture models?

Design models traced to requirements
How is this done today?

System of Systems

SoS Analysis

Full system (SoS) is field tested
How can MathWorks address the gap?
System Architecting & Technical Analysis

System Composer
- Import/construct architecture models
- Define architecture properties w/ profiles & stereotypes
- Analyze architectures
- Direct connection between architecture models & design models
Simulink Requirements
- Author/Import textual requirements
- Bi-directional traceability to architecture models, design models, & test cases
Verification & Validation

MATLAB, Simulink, Stateflow, Simscape
- Design multi-domain systems
- Early Verification by simulation, static analysis, & formal methods
Now let’s see it in action
## 1.4.2 Propulsion Power

**Summary:** Propulsion Power

**Description:**
- Gas Engine: 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 (338 kW) at 2,300 RPM for takeoff

**Rationale:**
- Gas Engine: 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 (338 kW) at 2,300 RPM for takeoff

**Keywords:**

**Revision information:**

**Links:**
- No links

**Comments:**


How to run the DeHaviland model:
Propulsion Power Subsystem

- Engine_Power_Nm_s

- apControls
- EnvBus
- enginePower

Requirements - UAS_reference_architecture

- Aircraft Capabilities
- Airworthiness
- Range
- Rain Conditions
- Power

Implemented

125%
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
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.
Create an instance model from this architecture model by flattening out all referenced models and their components. Such an instance model may be used for system-level analysis expressed as MATLAB functions.

Step 1: Select Stereotypes
Select the stereotypes to make available on the instance model

- UAVComponent
- SubsystemBudget

Step 2: Configure Analysis

Function
Analysis function:
- budgetRollUp

Function arguments (comma-separated):

>> budgetRollUp(instance)

Model Iteration
Mode: Bottom-up

Instance model name:
- UAS_reference_architecture_electric_budgetRollup

Don't see your profile? Profile Editor...
<table>
<thead>
<tr>
<th>Instance</th>
<th>Mass</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAS_reference_architecture_electric_budgetRollup</td>
<td>392.33</td>
<td>200614300</td>
</tr>
<tr>
<td>BVLOS Navigation</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ground Station</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Communication Box</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ground Station GPS interface</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>USB Serial Converter</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wireless Communication Subsystem</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GPS receiver</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Guidance and Navigation Computer</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Flight Commands</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Payload Computer</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vehicle</td>
<td>392.33</td>
<td>200614300</td>
</tr>
<tr>
<td>Communications Subsystem</td>
<td>2.63</td>
<td>56050</td>
</tr>
<tr>
<td>Automatic Dependent Surveillance-Broadcast</td>
<td>0.05</td>
<td>50000</td>
</tr>
<tr>
<td>C-Band Radio Modem</td>
<td>0.05</td>
<td>2000</td>
</tr>
<tr>
<td>KU-Band Radio TX/RX</td>
<td>2.5</td>
<td>50000</td>
</tr>
<tr>
<td>On-Board GPS</td>
<td>0.01</td>
<td>50</td>
</tr>
<tr>
<td>Radio RX PPM/PWM</td>
<td>0.02</td>
<td>1000</td>
</tr>
<tr>
<td>Electrical Subsystem</td>
<td>143.15</td>
<td>200355090</td>
</tr>
<tr>
<td>Actuator Power Subsystem</td>
<td>8</td>
<td>3000000</td>
</tr>
<tr>
<td>Power Distribution</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>Power Monitor</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Power Source</td>
<td>20</td>
<td>1000</td>
</tr>
<tr>
<td>SubsystemBudget</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Units</td>
<td>Value</td>
</tr>
<tr>
<td>Mass</td>
<td>kg</td>
<td>100</td>
</tr>
<tr>
<td>Power</td>
<td>mW</td>
<td>200,000,000</td>
</tr>
</tbody>
</table>
System Composer
Intuitively design system and software architectures
Simulink Requirements
Digital Thread from Requirements to Architecture and Design

Author requirements or view from external source

Identify gaps in architecture or design

Link requirements, architectures, design, code and test

Identify impact of requirement changes
System Composer
Tackle Architecture complexity with spotlight views
System Composer
System and software architectures connected to implementations in Simulink

Generate Simulink models from architecture components

Link Simulink models to architecture components
System Composer
Perform trade studies based on data driven analysis to optimize architectures

Add custom data

Create analysis model

Calculate mass roll-up data
Simulink: A Multi-Language Simulation Environment

Dynamic Systems

State Machines

Discrete-Event Systems

Physical Modeling

Programming Languages
Learn More

- Simulink Requirement Webpage
- System Composer Webpage
- System Modeling and Simulation Webpage

- Trial
Contact Us

508-647-7000

Monday - Friday

Customer Support ...... 08:30-17:30 ET
Technical Support ...... 08:30-20:00 ET