Model-Based Design for Digital Engineering: Impact and Directions

Richard Rovner

MATLAB EXPO FRANCE
Application Trends

- Autonomous
- Connectivity
- Electrification
Application Trends

- Autonomous
- Connectivity
- Electrification

Workflow Trends

- Systems Engineering & Design
- Modern Software Practices
- AI for System Development
Workflow Trends

Systems Engineering & Design
Modern Software Practices
AI for System Development
Workflow Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early

5. Apply standard software workflows
6. Design and simulate in the cloud

7. Design your system with AI
Workflow Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early
5. Apply standard software workflows
6. Design and simulate in the cloud
7. Design your system with AI
1. Automate everything

Modeling
- Simulation
- Analysis

Automation
- Coding
- Verification
Automate everything

From the simulation of multi-physics to the generation of industrial codes
Rémi Fayolle and Anthony Michel, Symbio
2. Scale to complex systems

**Components**

**Buses, Ports, and Connectors**

**Variant Manager**

**Third-Party Tool Integration**

**Architecture**
2 Scale to complex systems

Service-Oriented Architecture

System Composer

Full System

Components
Use automatic code generation

3700

Organizations use automatic code generation
Use automatic code generation
Prevent defects early
Prevent defects early

Find Defects Sooner

<table>
<thead>
<tr>
<th>Design</th>
<th>Test</th>
<th>Code</th>
<th>Certify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulink</td>
<td>Simulink</td>
<td>Polyspace Test Bug</td>
<td>DO Qualification Kit</td>
</tr>
<tr>
<td>Design</td>
<td>Check</td>
<td>Code Prover</td>
<td>IEC Certification Kit</td>
</tr>
<tr>
<td>Verifier</td>
<td>Coverage</td>
<td>Access</td>
<td>Simulink Code Inspector</td>
</tr>
<tr>
<td>HDL Verifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2023a</td>
<td>R2023a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Polyspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault</td>
<td>Test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyzer</td>
<td>Test</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
UL Certification of Battery Management System Software with Model-Based Design

The Saft Flex’ion Gen2
UL Certification of Battery Management System Software with Model-Based Design

The Saft Flex’ion Gen2
UL Certification of Battery Management System Software with Model-Based Design

The Saft Flex’ion Gen2
UL Certification of Battery Management System Software with Model-Based Design
UL Certification of Battery Management System Software with Model-Based Design

The Saft Flex’ion Gen2
UL Certification of Battery Management System Software with Model-Based Design

The Saft Flex’ion Gen2

2X
Workflow Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early

5. Apply standard software workflows
6. Design and simulate in the cloud

7. Design your system with AI
Apply standard software workflows

“Software is the language of automation.”
- Jensen Huang, co-founder and CEO of NVIDIA
Apply standard software workflows
Does your organization currently use a continuous integration (CI) system? (select one)
Apply standard software workflows
Apply standard software workflows

**Workflow Steps**

The workflow consists of the following steps (Figure 4):

1. Trigger a pipeline in GitLab and observe that the Verify and Build stage passes.
2. Detect a test-case failure in GitLab CI pipeline and create an issue to track.
3. Reproduce the issue on our desktop MATLAB.
4. Fix the issue in the model.
5. Test locally to ensure the test case passes.
6. Review the changes on the testing branch.
7. Commit changes to Git and trigger the CI pipeline in GitLab.
From Scripted Pipelines to Process Advisor
From Scripted Pipelines to Process Advisor

Jenkins • git

MATLAB® & SIMULINK®

AUTOSAR
“Bring everything into MATLAB.”

- Martin Römpert, Continental Automotive Technologies GmbH
Leverage the digital thread
Identify stale tests
Interact with the model

Process Advisor

From Scripted Pipelines to Process Advisor
From Scripted Pipelines to Process Advisor
From Scripted Pipelines to Process Advisor

Process Advisor

500 interfaces
1,000 components
100 compositions
Design and simulate in the cloud
Design and simulate in the cloud

```matlab
for i = 1:10000
    in(i) = Simulink.SimulationInput(my_model);
in(i) = setVariable(my_var, i);
end
out = parsim(in);
```
Design and simulate in the cloud

Global Combat Air Programme


Future Combat Air System

Source: Model of the Future Air Combat System at the Paris-Le Bourget 2019 Airshow, by Joxe73, licensed under CC-BY-SA 4.0 / background logos blurred from original
Application Trends

Autonomous
Connectivity
Electrification

Workflow Trends

Systems Engineering & Design
Modern Software Practices
AI for System Development
Workflow Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early
5. Apply standard software workflows
6. Design and simulate in the cloud
7. Design your system with AI
Design your system with AI

Integrating AI into Model-Based Design

- Requirements
- Functionality and Architecture
- Design
- Implementation
- System Integration and Test

- **System Architecture**
  - Behavior Models
  - Functional Spec

- **Environment Model**

- **Subsystem Models**
  - Physics Based AI & Data Driven

- **Algorithms**
  - Models, Code, AI

- **System**
  - CPU • GPU
  - DSP • FPGA
  - IEC 61131 for PLCs
  - Microcontroller
  - Software Component
  - App • Container
  - Microservice

- **Continuous Testing, Verification, and Validation**
  - Requirements verification
  - Early design verification
  - Virtual integration testing (SIL/PIL/HIL)
  - Physical system testing

- **Digital Thread**

- **Physical Hardware**

- **Component and System Acceptance Testing**

- **Real Environment**
7 Design your system with AI

Masterclass: “AI and Model-Based Design”
AI at the service of systems simulation
Moubarak Gado, MathWorks
Design your system with AI

AI Reference Examples

- Predictive Maintenance
- Hyperspectral Imaging
- Signal Processing
- Robotic Control
- Lidar Processing
- Radar Processing
- Wireless Communications
- Automated Driving
- Visual Inspection
- Reinforcement Learning
- Audio
- Medical Imaging
Design your system with AI

AI Reference Examples

- Predictive Maintenance
- Hyperspectral Imaging
- Signal Processing
- Robotic Control
- Lidar Processing
- Radar Processing
- Wireless Communications
- Automated Driving
- Visual Inspection
- Reinforcement Learning
- Audio
- Medical Imaging

CPUs, GPUs, and FPGA, ASIC, PLC are often used to support AI applications.
Simulates Hardware Sensors with Deep Neural Networks
Simulates Hardware Sensors with Deep Neural Networks

TRAINING

Machine Learning Algorithm

Trained Neural Net

TRANSFER

Net object of trained neural net

Fixed point Simulink model of trained neural net

IMPORT

CONVERT & "SCALE"

SOFTWARE INTEGRATION

TARGET HARDWARE

Powertrain Controller

Using Deep Learning Toolbox

Using custom library and Fixed-Point Designer
“We are already using the automated workflow we created with MATLAB and Simulink for other use cases ... small adaptations to support deployment on two different powertrain controllers, and the workflow is also applicable to other types of deep learning models such as gated recurrent units and fully connected neural networks ... we committed fewer errors in creating the model and the code.”

- Katja Deuschi, AI Developer, Mercedes-Benz
Simulates Hardware Sensors with Deep Neural Networks

6X
Workflow Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early
5. Apply standard software workflows
6. Design and simulate in the cloud
7. Design your system with AI
Application Trends

Autonomous

Connectivity

Electrification

Workflow Trends

Systems Engineering & Design

Modern Software Practices

AI for System Development
Deliver autonomous systems

Braking and Steering

Sensor Fusion and Tracking

Computer Vision Radar, Lidar

Road Network Design
Unmanned Aerial Vehicle

Autonomous Underwater Vehicle

Ground Robot

Industrial Robot
Application Trends

- Autonomous
- Connectivity
- Electrification

Workflow Trends

- Systems Engineering & Design
- Modern Software Practices
- AI for System Development
Application Trends

- Autonomous
- Connectivity
- Electrification

Workflow Trends

- Systems Engineering & Design
- Modern Software Practices
- AI for System Development
5G Standard
"One of MATLAB’s differentiators is its vertical products like 5G Toolbox. We use that toolbox to generate datasets for testing algorithms. We have not been able to find that capability in other software suites."

- Christopher Brinton, Professor of Electrical and Computer Engineering, Purdue University
6G Technology Implications

- Artificial Intelligence
- Joint Communications and Sensing
- Reconfigurable Intelligent Surfaces
- Non-Terrestrial Networks (NTNs)
- Physical Layer Design
- Extreme Data Rates and Higher Frequencies
Wireless Trends – AI in Wireless

Wireless challenges
- Hard-to-model problems
- Computational infeasibility of optimal solution
- Efficient modem parameter optimization
- Dealing with non-linearity

AI strengths
- Determining appropriate representations for hard-to-model problems
- Finding near-ideal and computationally realizable solutions
- Modeling non-linear functions

Applying AI to solve difficult wireless challenges
Deep wireless domain knowledge is required to optimally use AI capabilities
Application Trends

Autonomous

Connectivity

Electrification

Workflow Trends

Systems Engineering & Design

Modern Software Practices

AI for System Development
Application Trends

- Autonomous
- Connectivity
- Electrification

Workflow Trends

- Systems Engineering & Design
- Modern Software Practices
- AI for System Development
Electric Vehicles

Accelerate the development and implementation of automotive BMS features
Vincent Martinez and Léa Pitault, NXP Semiconductors

Electric Motors

Battery Packs

Full Vehicle Models

Aerial Vehicles
Green Energy
Enables the Global Energy Transition

1. Engineer solutions in solar, biomass, hydrogen, wind
2. Retrofit or upgrade infrastructure
3. Strengthen electrical grid
Swimlane Engineering
When the organization shapes development

<table>
<thead>
<tr>
<th>Development – Time</th>
<th>Runtime (HW)</th>
<th>Test / Integration - Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>Language A</td>
<td>Compiler A</td>
</tr>
<tr>
<td></td>
<td>Binary</td>
<td>FPGA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team B</td>
<td>Language B</td>
<td>Compiler B</td>
</tr>
<tr>
<td></td>
<td>Binary</td>
<td>DSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardware-in-the-Loop (HiL)</td>
</tr>
<tr>
<td>Team C</td>
<td>Language C</td>
<td>Simulation Tool</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PC</td>
</tr>
</tbody>
</table>
### Centralized Engineering Ecosystem
When development extends across the organization

<table>
<thead>
<tr>
<th>Development – Time</th>
<th>Runtime (HW)</th>
<th>Integration - Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>Binary</td>
<td>Hardware-in-the-Loop (HiL)</td>
</tr>
<tr>
<td>Team B</td>
<td>FPGA</td>
<td>=</td>
</tr>
<tr>
<td>Team C</td>
<td>DSP</td>
<td>Offline System Simulation</td>
</tr>
</tbody>
</table>

Model-in-the-Loop (MiL)

Enables the Global Energy Transition
Enables the Global Energy Transition

Putting everything together
A Simulink based digital twin lets us analyze and test our system early on
Enables the Global Energy Transition

Reference Project Lines
Standardization for individual Customer Projects
Application Trends

Autonomous

Connectivity

Electrification

Workflow Trends

Systems Engineering & Design

Modern Software Practices

AI for System Development
Application Trends

Workflow Trends

Autonomous

Connectivity

Electrification

Systems Engineering & Design

Modern Software Practices

AI for System Development
Application Trends

1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early
5. Apply standard software workflows
6. Design and simulate in the cloud
7. Design your system with AI
Thank you