

Simulink as Your **Enterprise Simulation Platform**

MATLAB EXPO 2017



Stephan van Beek
Manager, Applications Engineering Group

Why simulation?



Hyperloop



Hyperloop is a revolutionary concept for a new type of rapid, low-energy transport system using magnetic levitation, which would be capable of transporting people and goods through extremely low-pressure tubes at speeds of up to Mach .98, or 750 mph/1,200 kmh.

Enterprise Simulation Platform

Enterprise: connects complete product development process



Research



Development

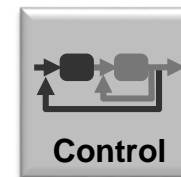


Production

Simulation: evaluating system behavior through computation



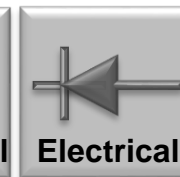
Platform: connects all relevant domains for modelling and simulation



Control



Mechanical



Electrical

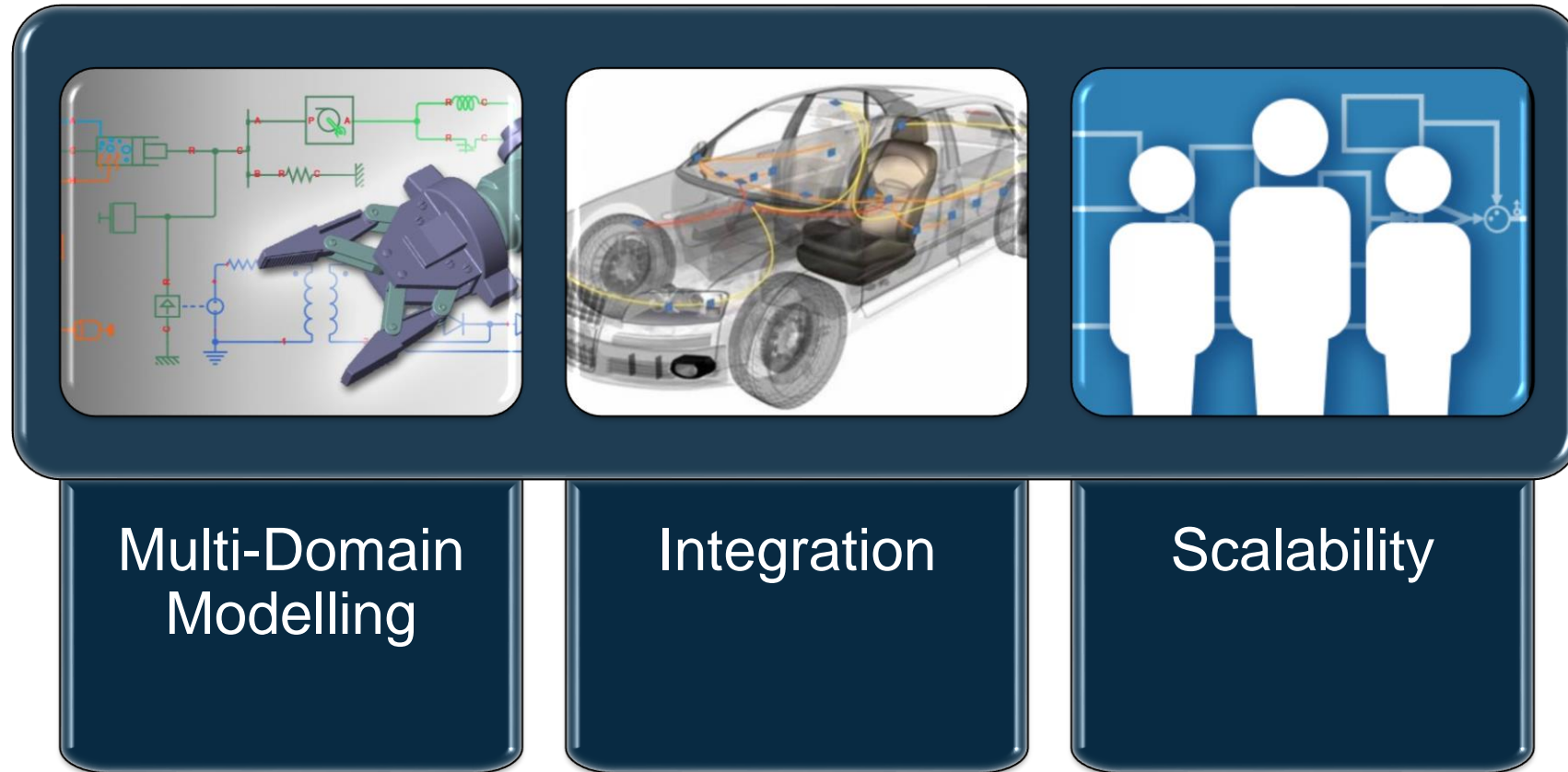


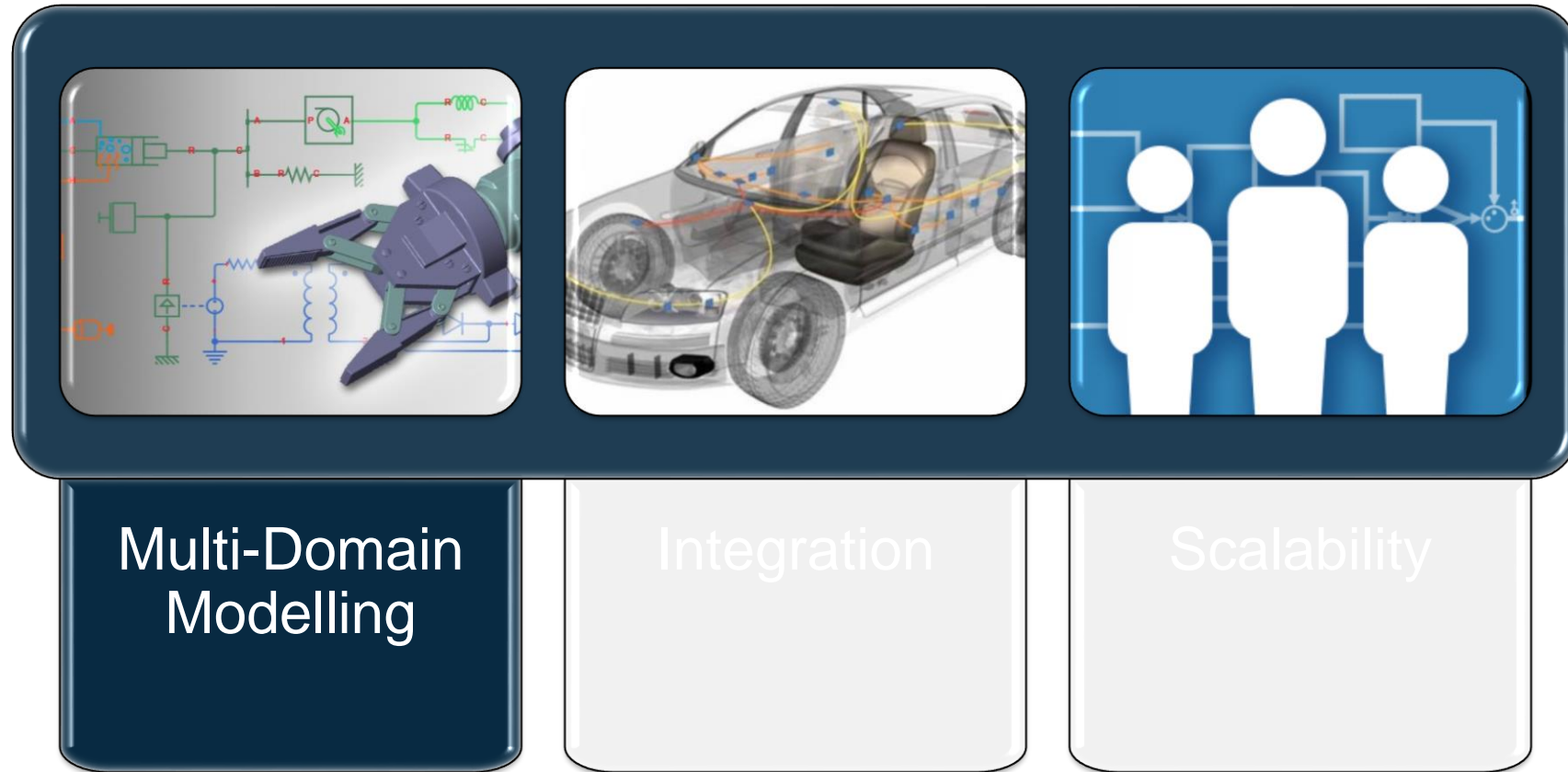
FE Tools



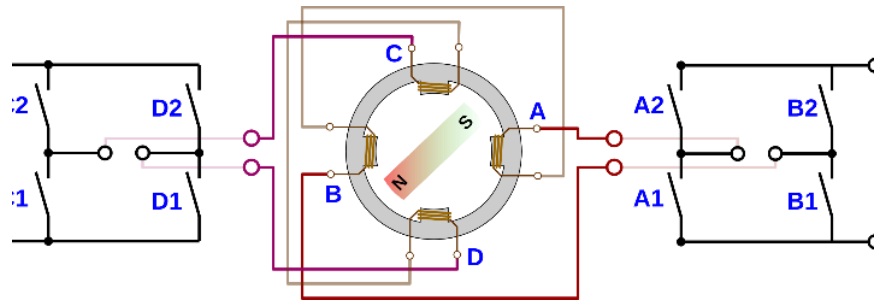
EDA Tools

Enterprise Simulation Platform Enablers





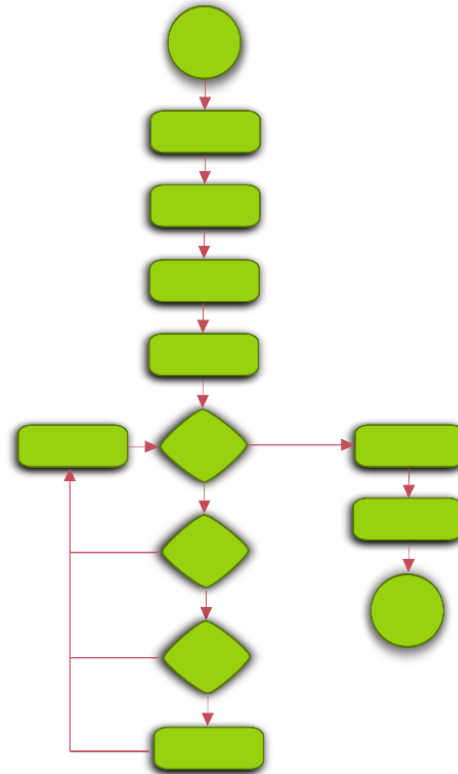
Multi-Domain Modelling in Simulink



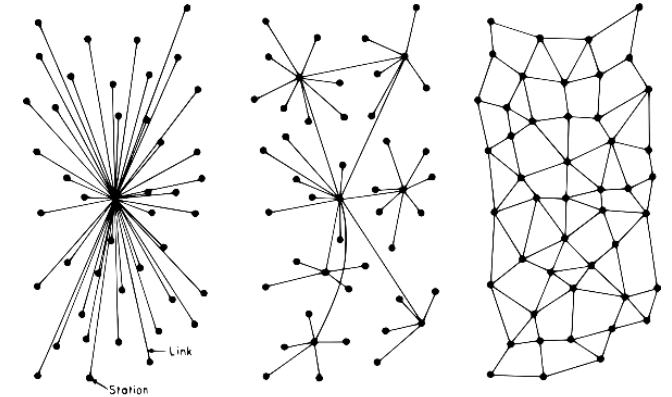
Dynamic Systems



Physical Modelling



State Machines

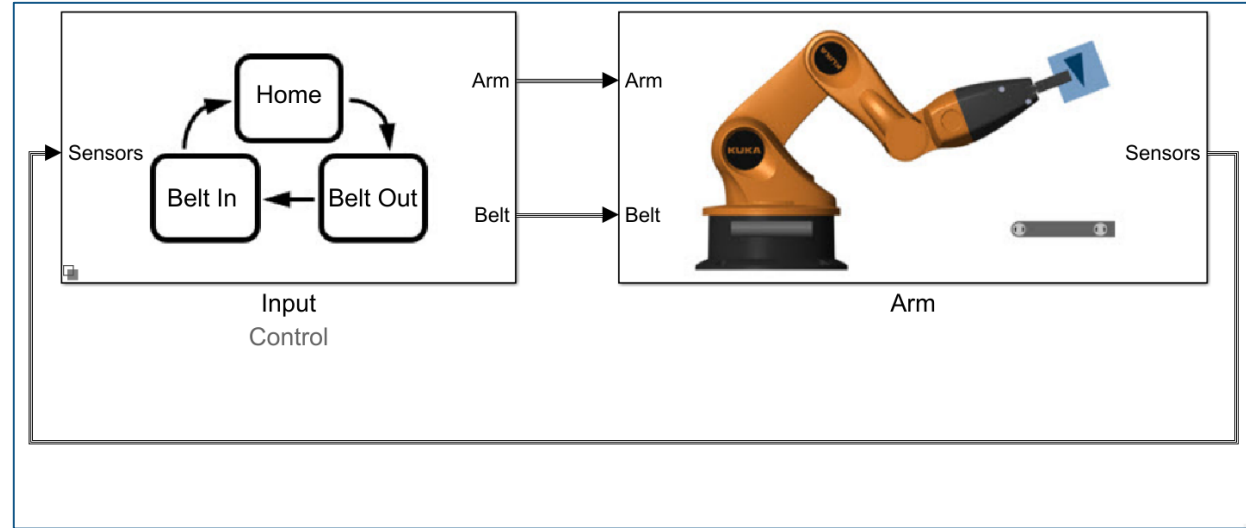
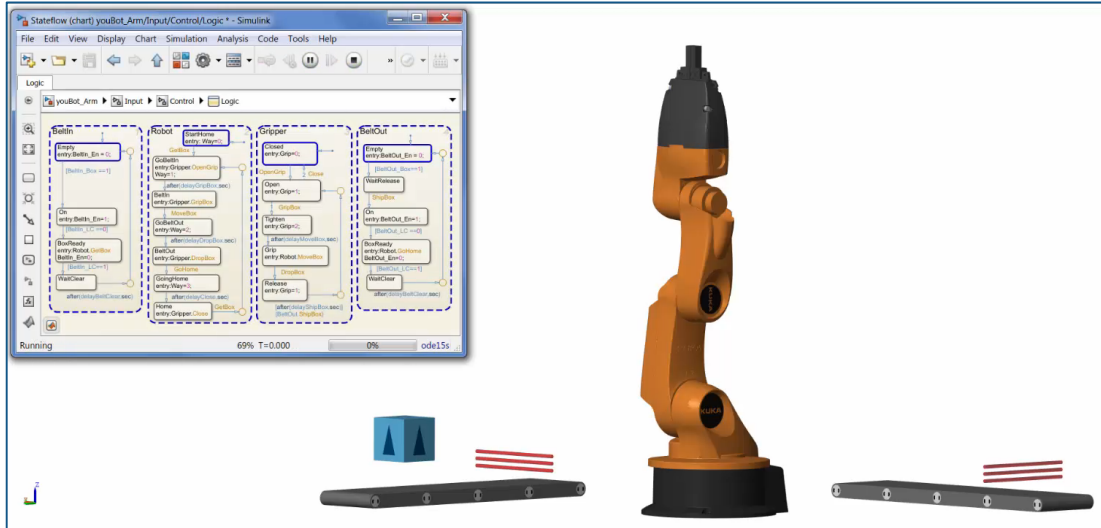


Discrete-Event Systems

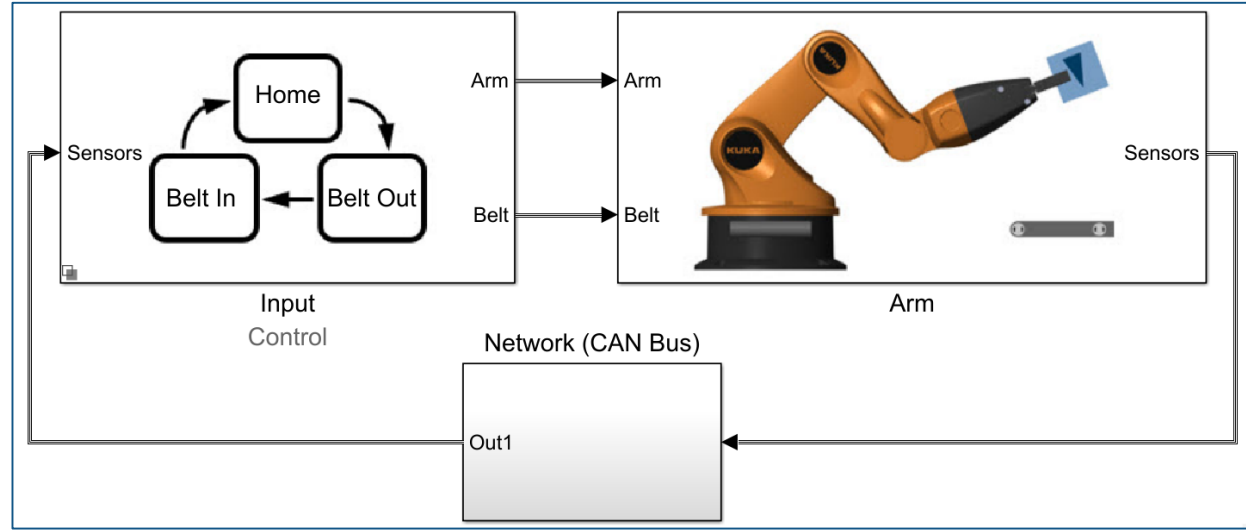
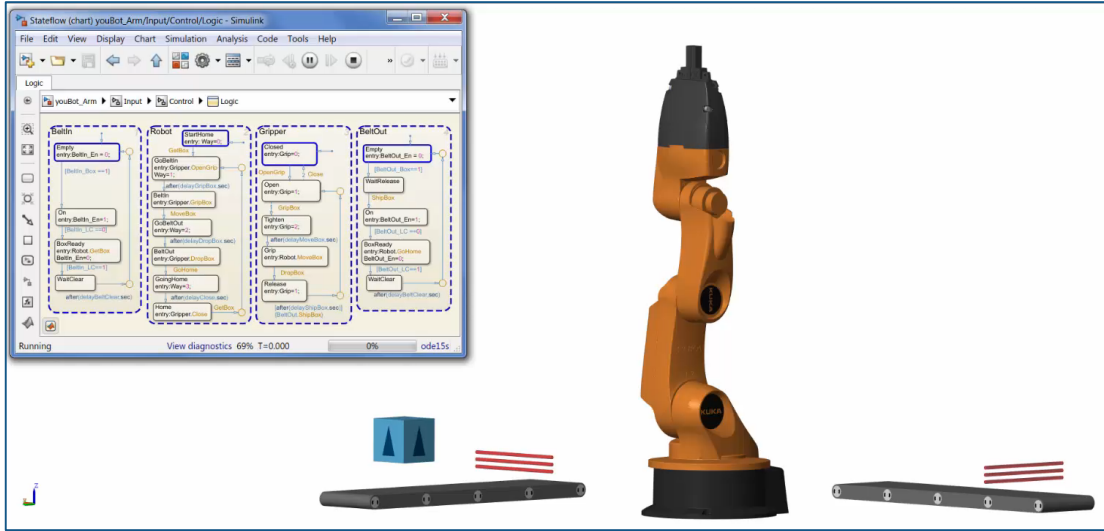


Function and Object-Oriented

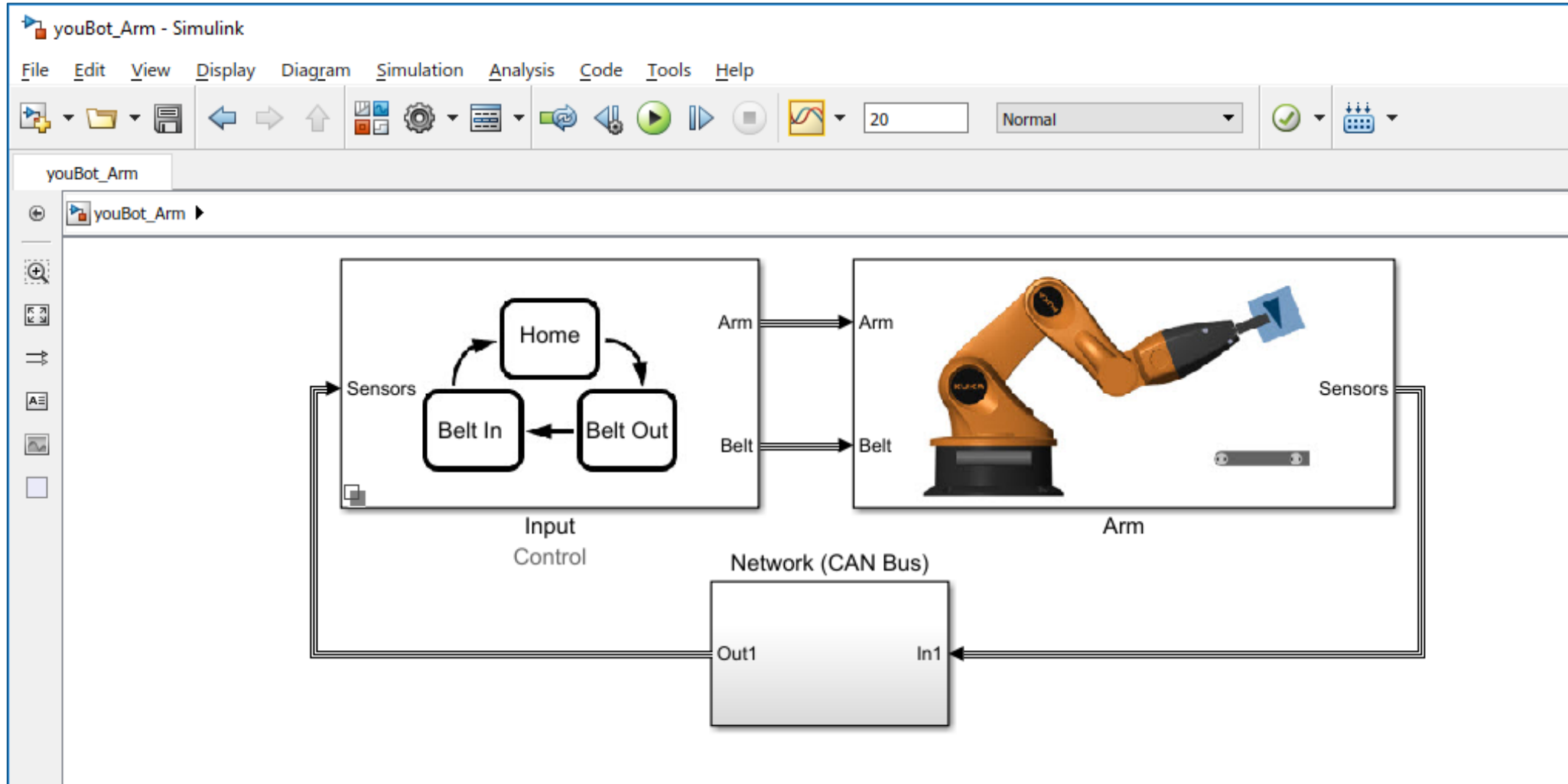
Need for Multi-Domain Simulation



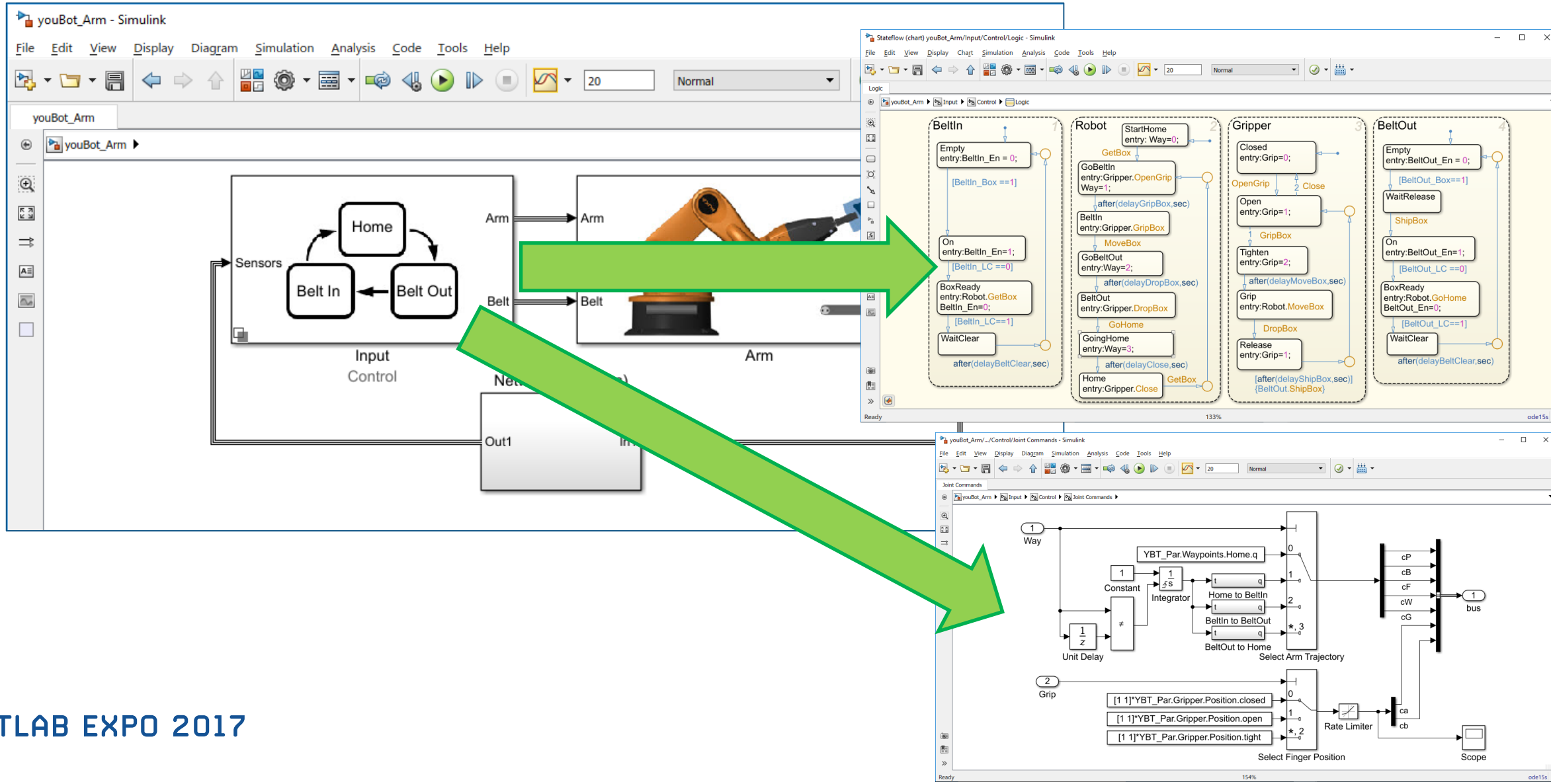
Need for More Multi-Domain Simulation



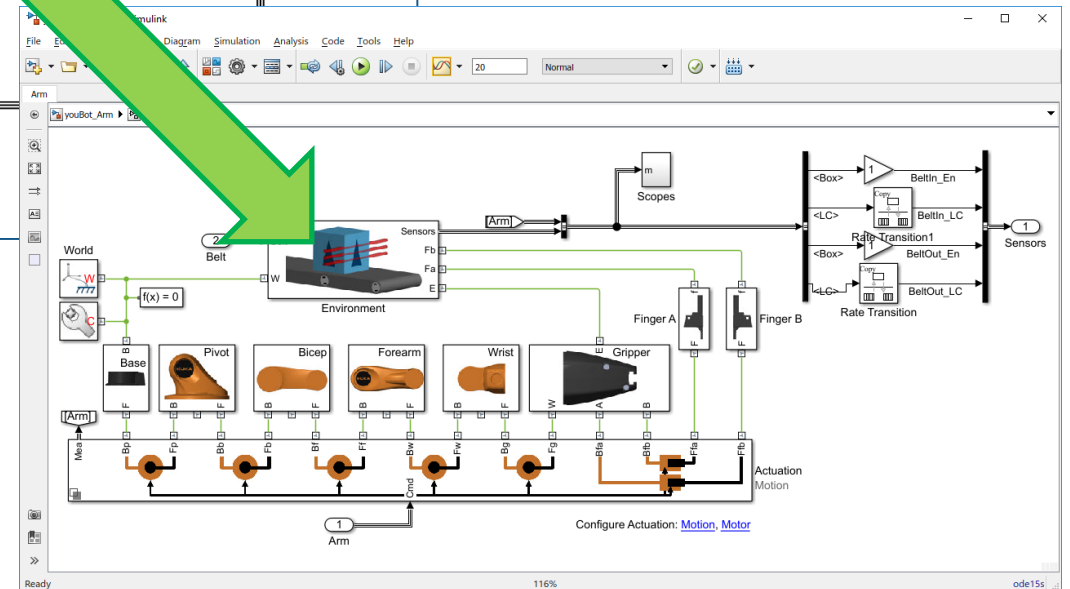
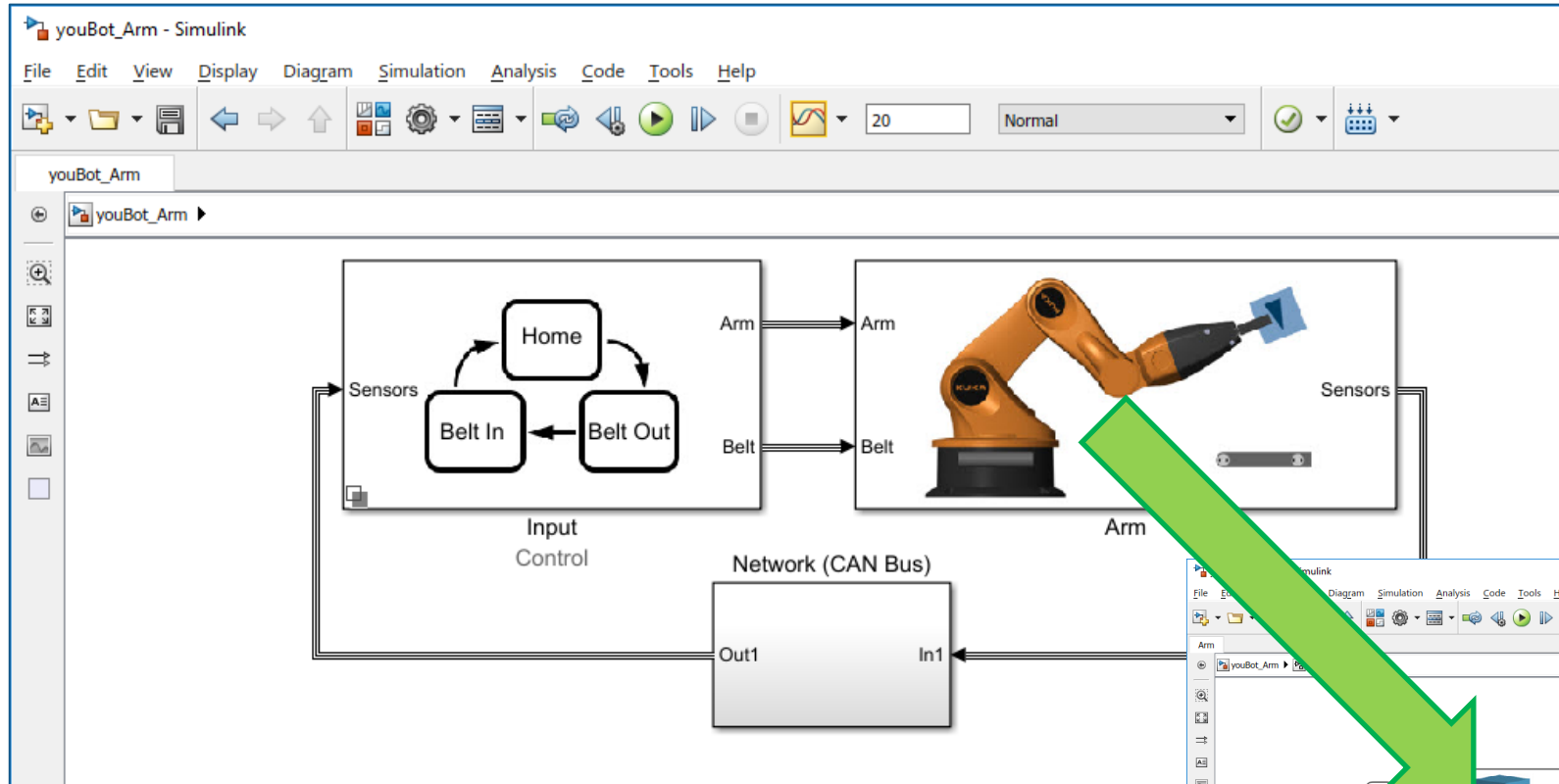
Multi-Domain Model



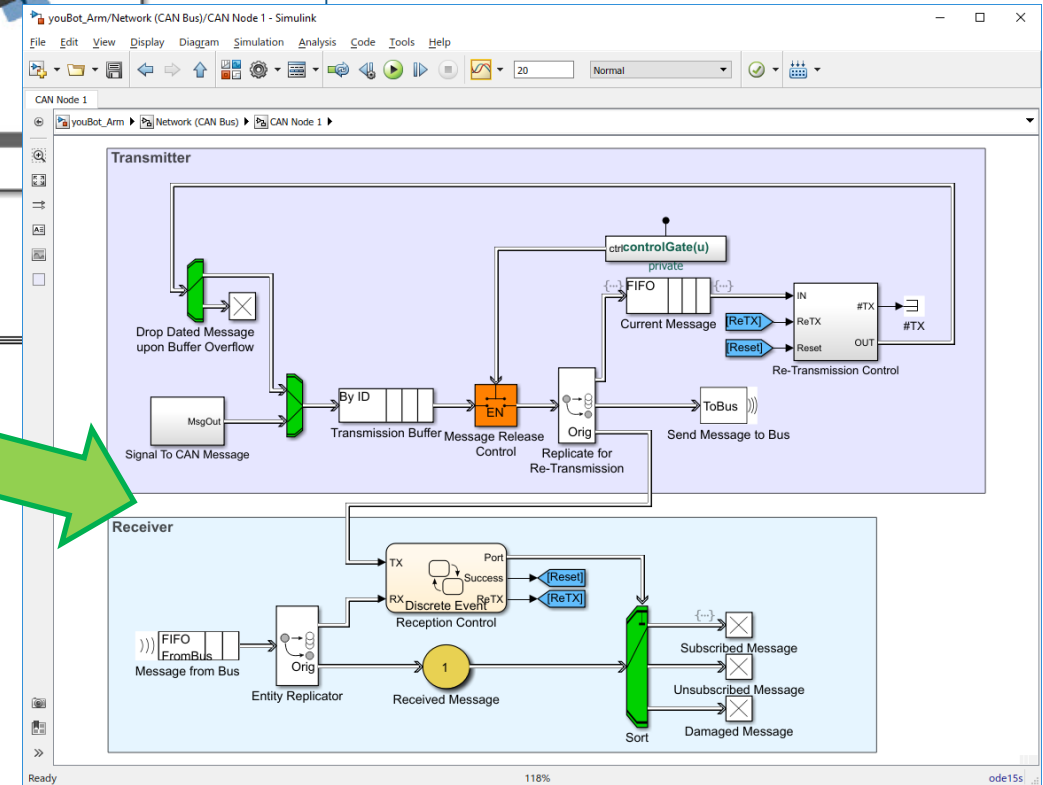
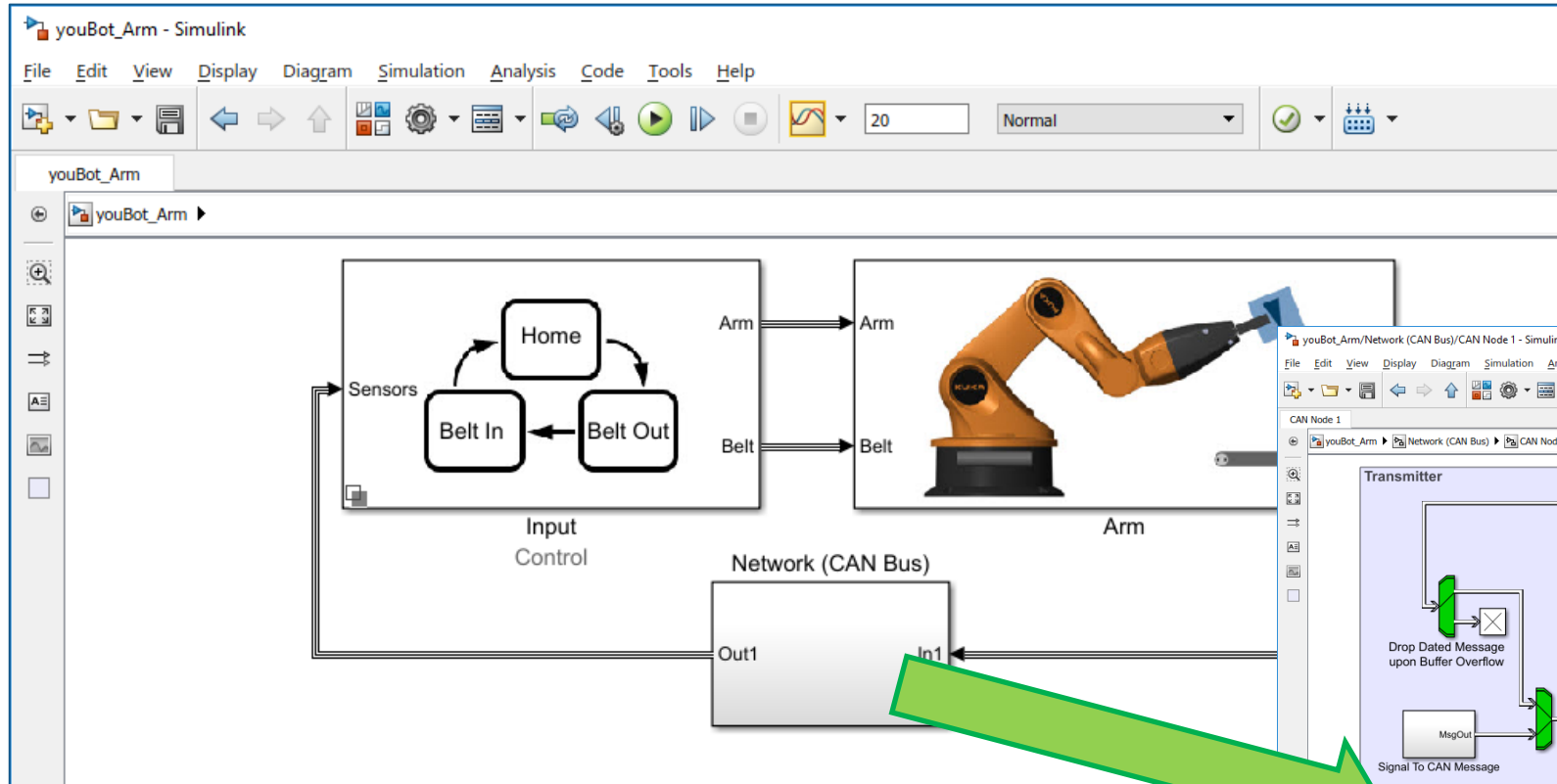
State Charts and System Dynamics



Physical Modeling



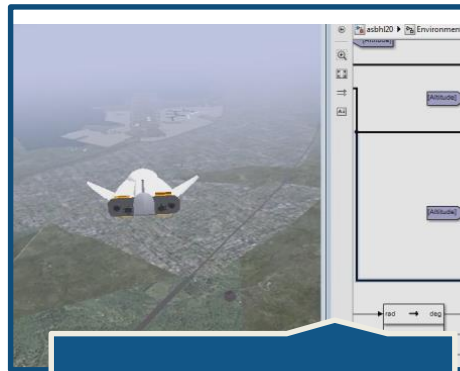
Discrete-Event Modeling



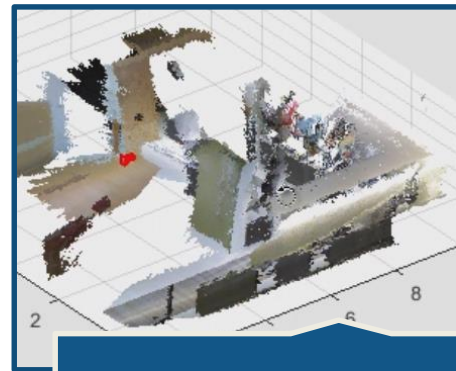


Domain-Specific Extensions

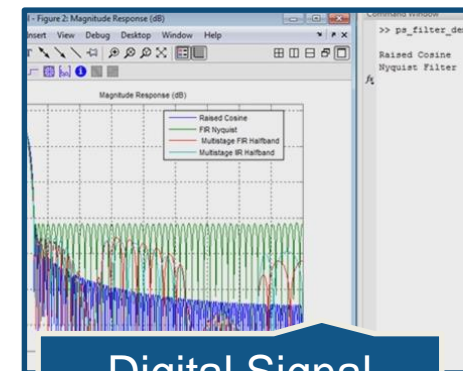
Simulink has numerous domain-specific capabilities, for example:



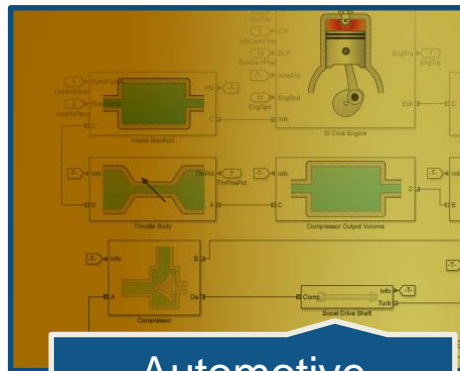
Aerospace



Computer Vision



Digital Signal Processing



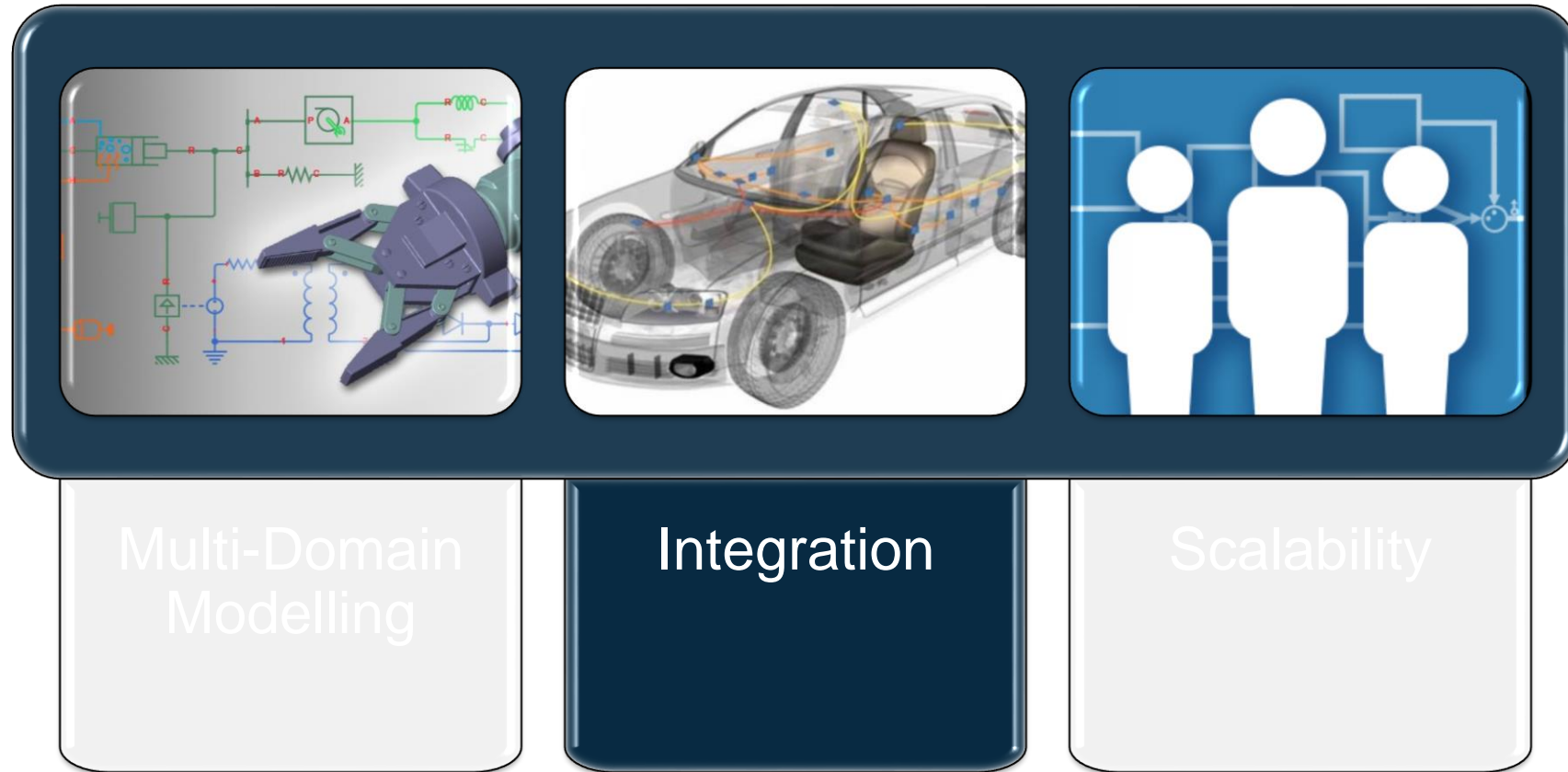
Automotive Powertrains

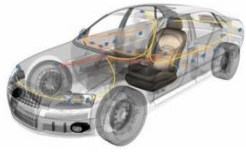


Robotic Applications



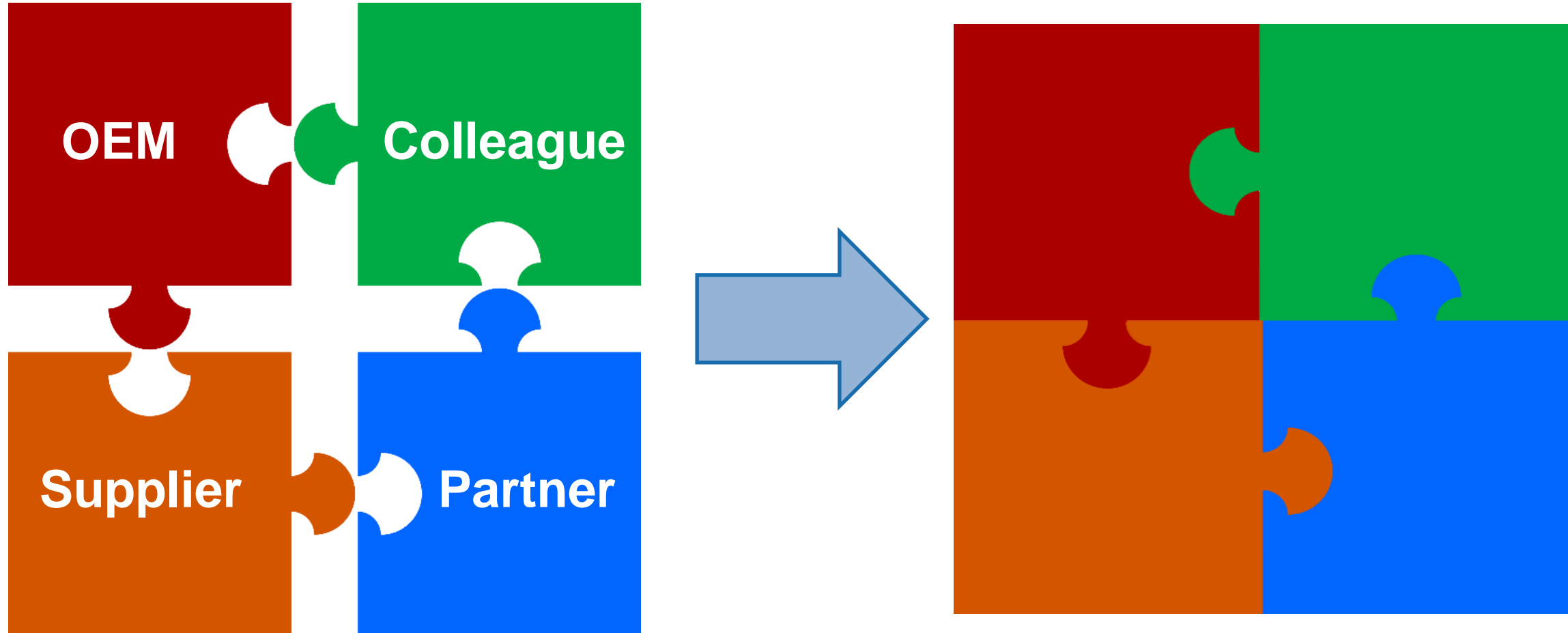
+More

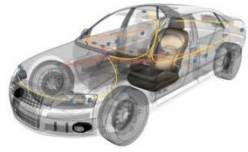




Integration Challenges

Your IP exists in many forms and in many locations, making integration difficult



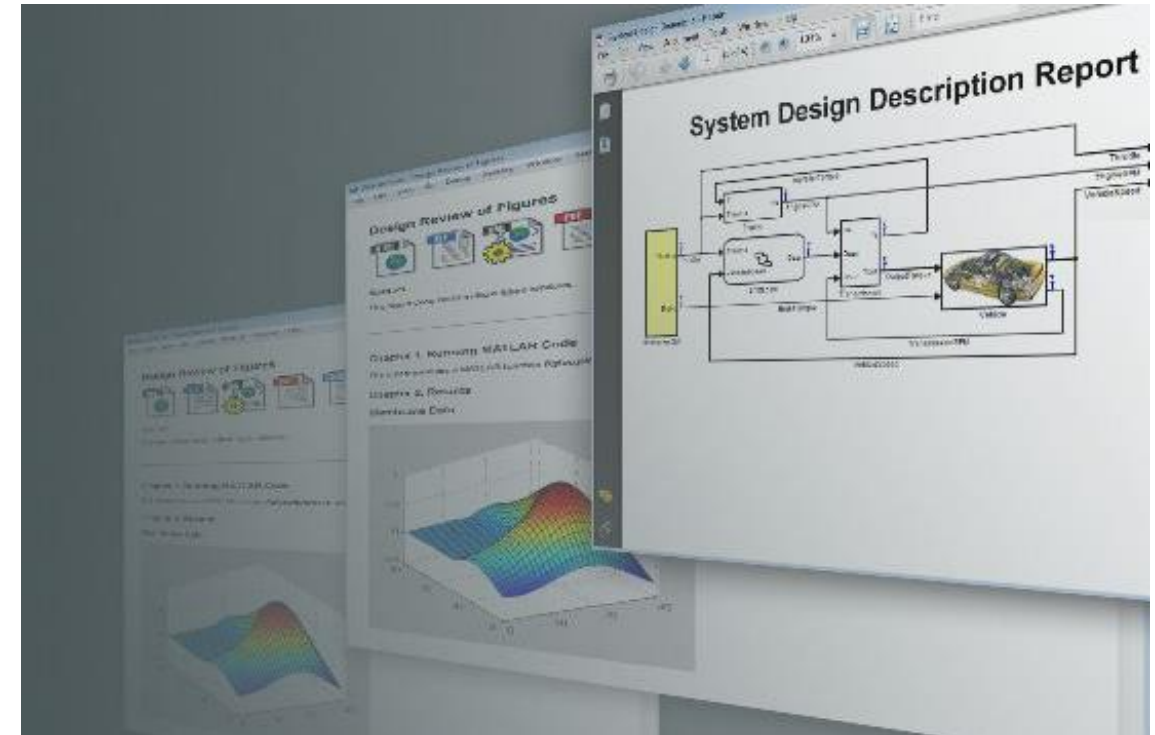


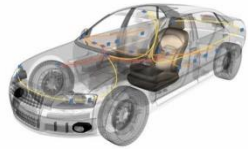
Integrating by Sharing Models

Quick File Packaging

Model Protection (IP Management)

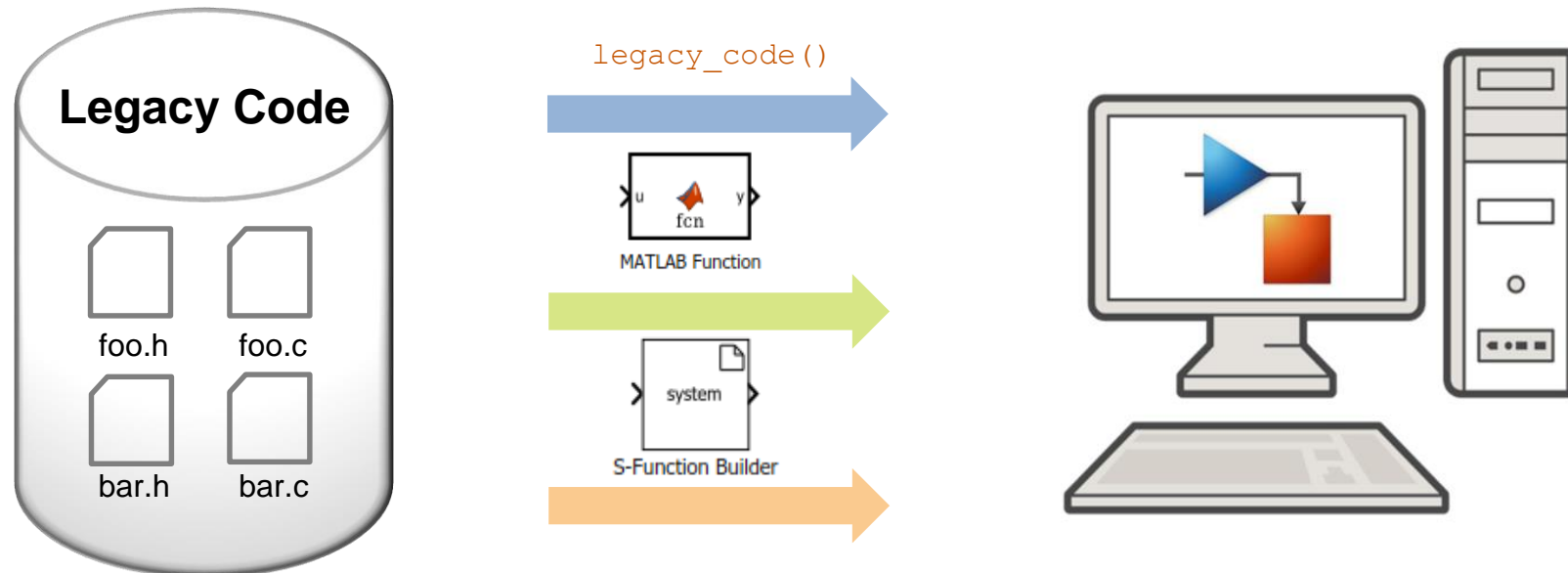
Reporting and Documentation

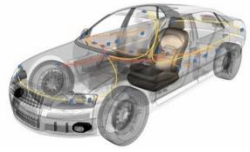




Integrating Your Code

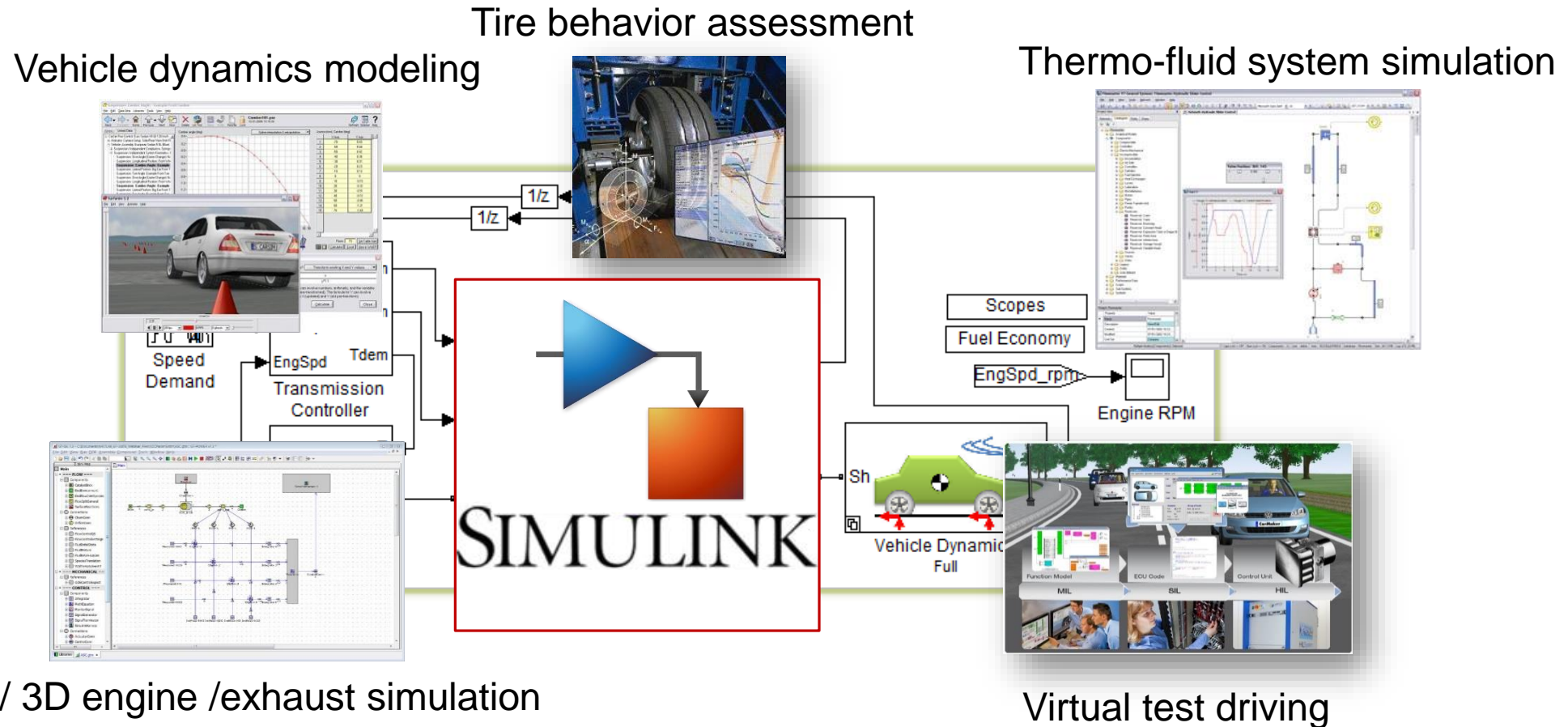
Multiple ways to reuse your legacy code with Simulink



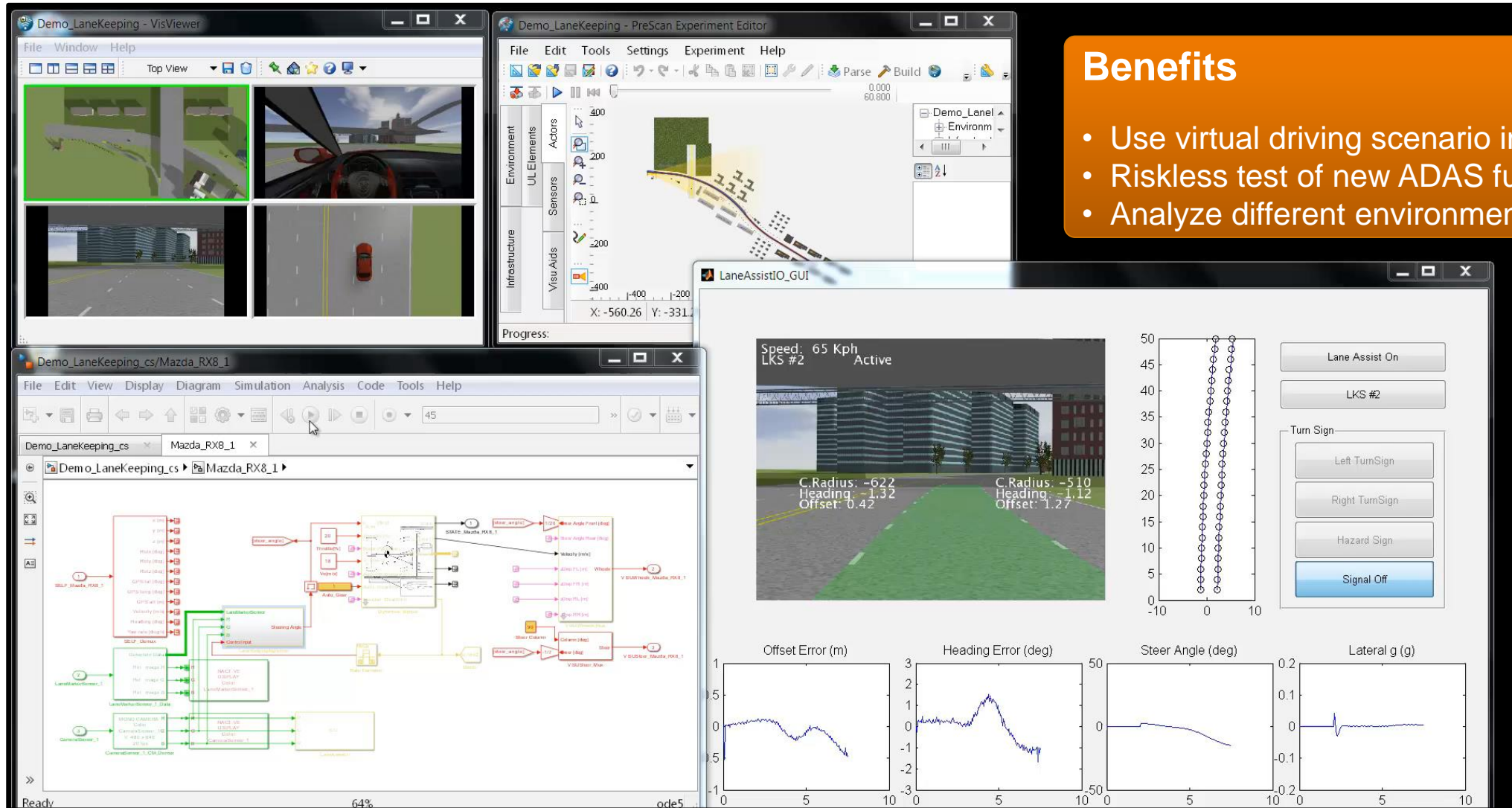


Integrating Third-Party Simulation Tools

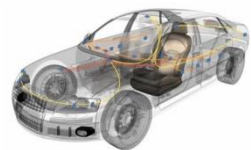
Mature and extensive APIs for third-party tool integration



ADAS System Level Simulation – Lane Keeping Support at TASS

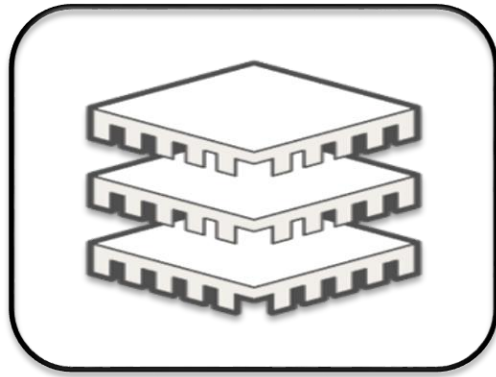


- ## Benefits
- Use virtual driving scenario instead of real car
 - Riskless test of new ADAS functions
 - Analyze different environmental conditions





Scalability Challenges

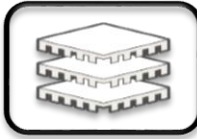


Performance



Team
Workflows

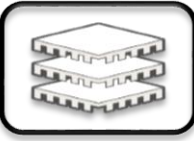
Performance Scalability



Easy scalability to multicore or cluster/cloud computation environment

The screenshot shows the MATLAB R2017a Command Window with the following text:

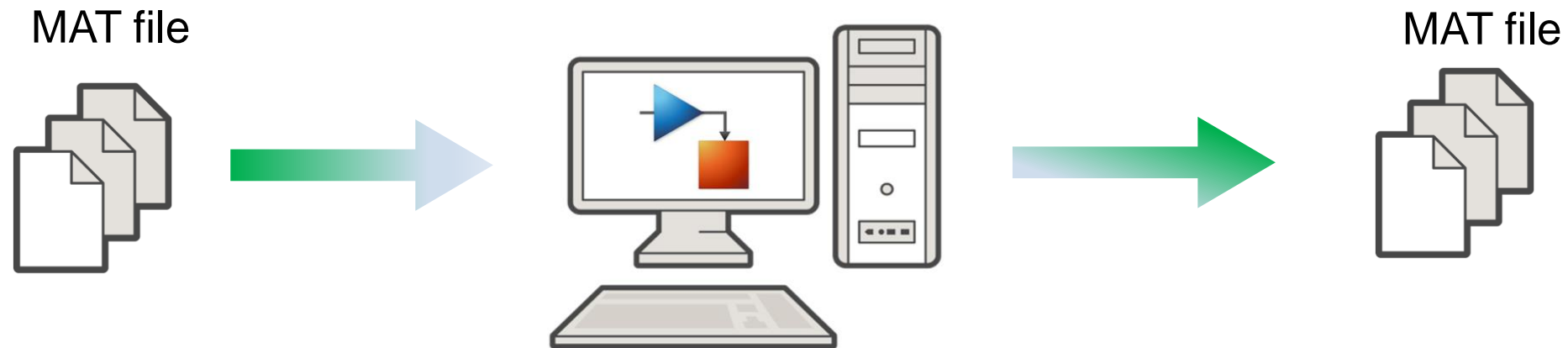
```
Completed 1827 of 10000 simulation runs
Completed 1828 of 10000 simulation runs
Completed 1829 of 10000 simulation runs
Completed 1830 of 10000 simulation runs
Completed 1831 of 10000 simulation runs
Completed 1832 of 10000 simulation runs
Completed 1833 of 10000 simulation runs
Completed 1834 of 10000 simulation runs
Completed 1835 of 10000 simulation runs
Completed 1836 of 10000 simulation runs
Completed 1837 of 10000 simulation runs
Completed 1838 of 10000 simulation runs
Completed 1839 of 10000 simulation runs
Completed 1840 of 10000 simulation runs
Completed 1841 of 10000 simulation runs
Completed 1842 of 10000 simulation runs
Completed 1843 of 10000 simulation runs
Completed 1844 of 10000 simulation runs
```



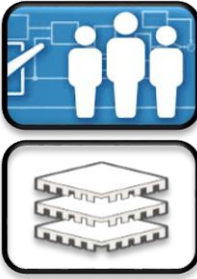
Performance Scalability

Big data workflow

- Processing large amount of simulation inputs / outputs

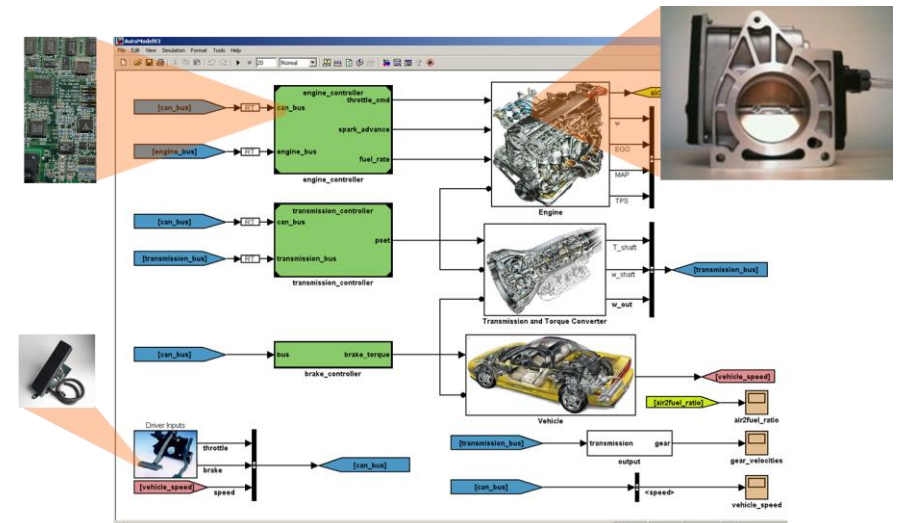
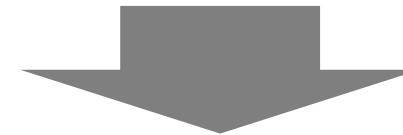
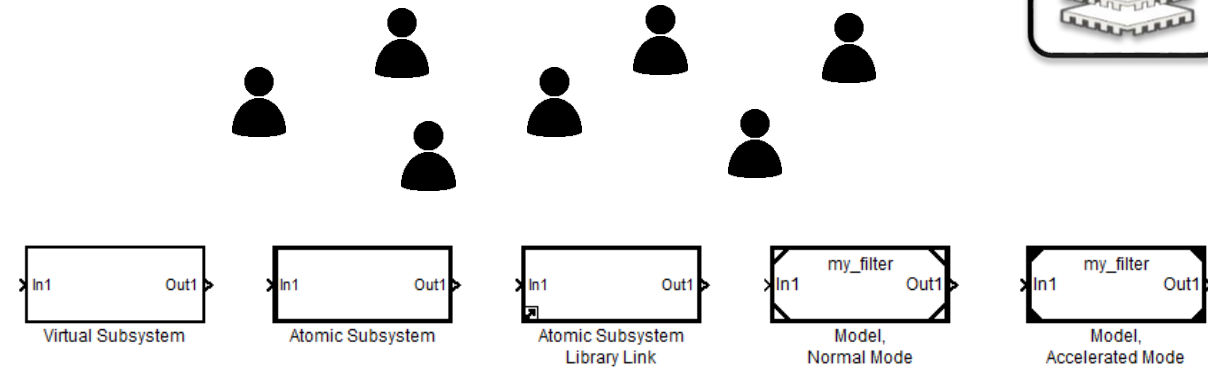


Complex Design Development through Componentization



- Improving performance
 - Incremental loading and code generation
 - Simulation speed
 - Memory usage

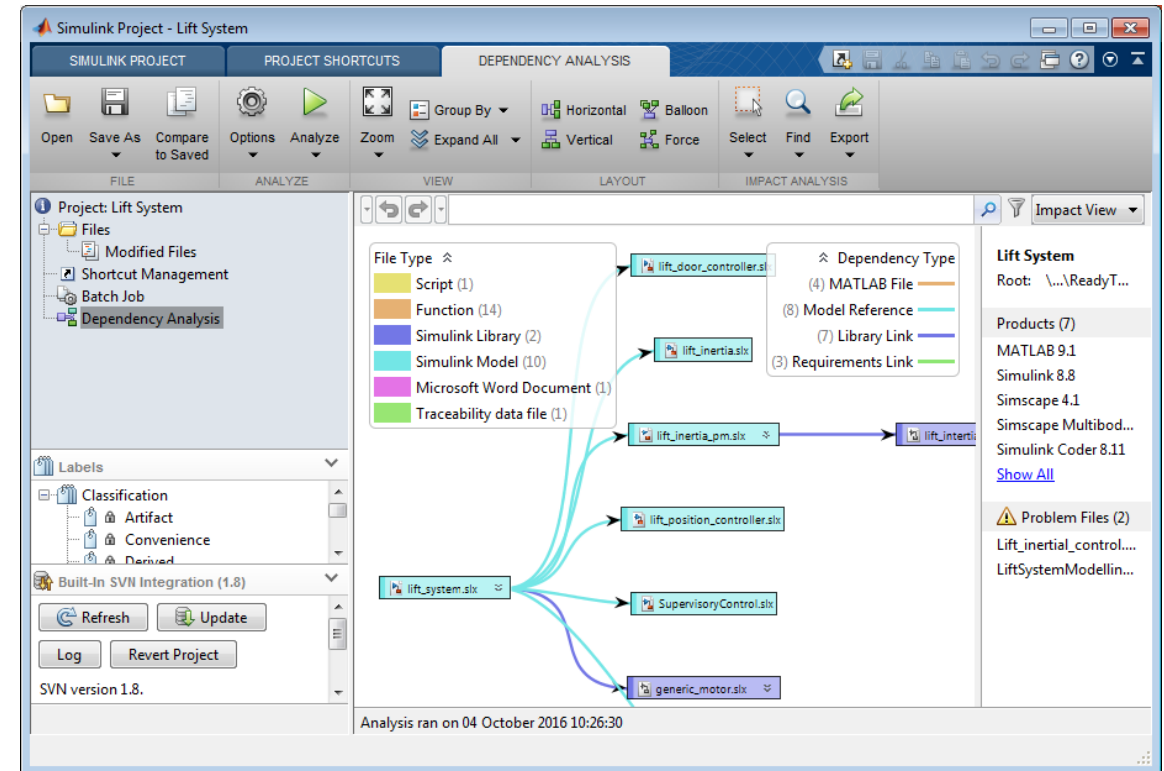
- Supporting team workflows
 - Faster modular development
 - More effective verification
 - Increased reusability



Capabilities Enabling Team Workflows



- Source control
- Design comparison and merging
- Dependency analysis
- Task automation



Source Control Integrations



Microsoft Team Foundation Server (TFS) integration available now from MathWorks File Exchange



Products Solutions Academia Support **Community** Events

File Exchange

TFS Version Control Integration

by [Jasper Schneider](#)

17 May 2016 (Updated 26 May 2016)

TFS Version Control integration in MATLAB and Simulink

[Watching this File](#)



Integrating Work from Different Engineers via Merge



The screenshot shows the Simulink Project - Elevator System interface. The main window displays a table of project files with the following columns: Name, SVN, Classification, Status, and Revision. The table is sorted by Name. The file 'lift_position_controller' is highlighted, showing a red exclamation mark in the SVN column, indicating a conflict or error. The Status column shows a green checkmark and a document icon for this file.

Name	SVN	Classification	Status	Revision
BatchJobs	●	None	✓	5
Components	●	Design	✓	28
Lift Door Controller	●	Design	✓	2
Lift Doors	●	Design	✓	13
Lift Inertia	●	None	✓	3
Lift Position Controller	●	Design	✓	28
lift_position_controller	●	Design	✓	28
Lift Supervisory Controller	●	Design	✓	27
Motors	●	Design	✓	13
Visualization	●	Design	✓	2
Documents	●	None	✓	2
SystemLevelModels	●	None	✓	12
Tests	●	Test	✓	8
Utilities	●	None	✓	11

- Supports concurrent engineering
- Lets you concentrate on design



Dependency Analysis – Modular Development

Simulink Project - Elevator System

SIMULINK PROJECT PROJECT SHORTCUTS

MANAGE DOCUMENTATION ENVIRONMENT TOP LEVEL MODELS UTILITIES

Project: Elevator System

Files

- Shortcut Management
- Batch Job
- Dependency Analysis

Name	Path	Status	Classification
Lift	\$\Tests	✓	Test
Lift Door Controller	Components	✓	Design
Lift Door Controller	Tests	✓	Test
Lift Doors	Components	✓	Design
Lift Doors	Tests	✓	Test
Lift Inertia	Components	✓	None
Lift Motor	Tests	✓	Test
Lift Position Controller	Components	✓	Design
Lift Position Controller	Tests	✓	Test
Lift Supervisory Controller	Components	✓	Design
Lift Supervisory Controller	Tests	✓	Test
Motors	Components	✓	Design
SystemLevelModels	\$\	✓	None
Tests	\$\	✓	Test
Utilities	\$\	✓	None
Visualization	Components	✓	Design
Visualization	Tests	✓	Test
basic_animation.slx	Components\Visualization	✓	Design
ElevatorTemplate.slx	Utilities	✓	Other
exportToR2016a.m	BatchJobs	✓	Design
generateBillOfMaterials.m	BatchJobs	✓	Design
generateICD.m	Utilities	✓	Design
generic_motor.slx	Components\Motors	✓	Design
history.m	Utilities	✓	Design
lift_door.req	Components\Lift Doors	✓	Design
lift_door.slx	Components\Lift Doors	✓	Design
lift_door_controller.slx	Components\Lift Door Controller	✓	Design

Labels

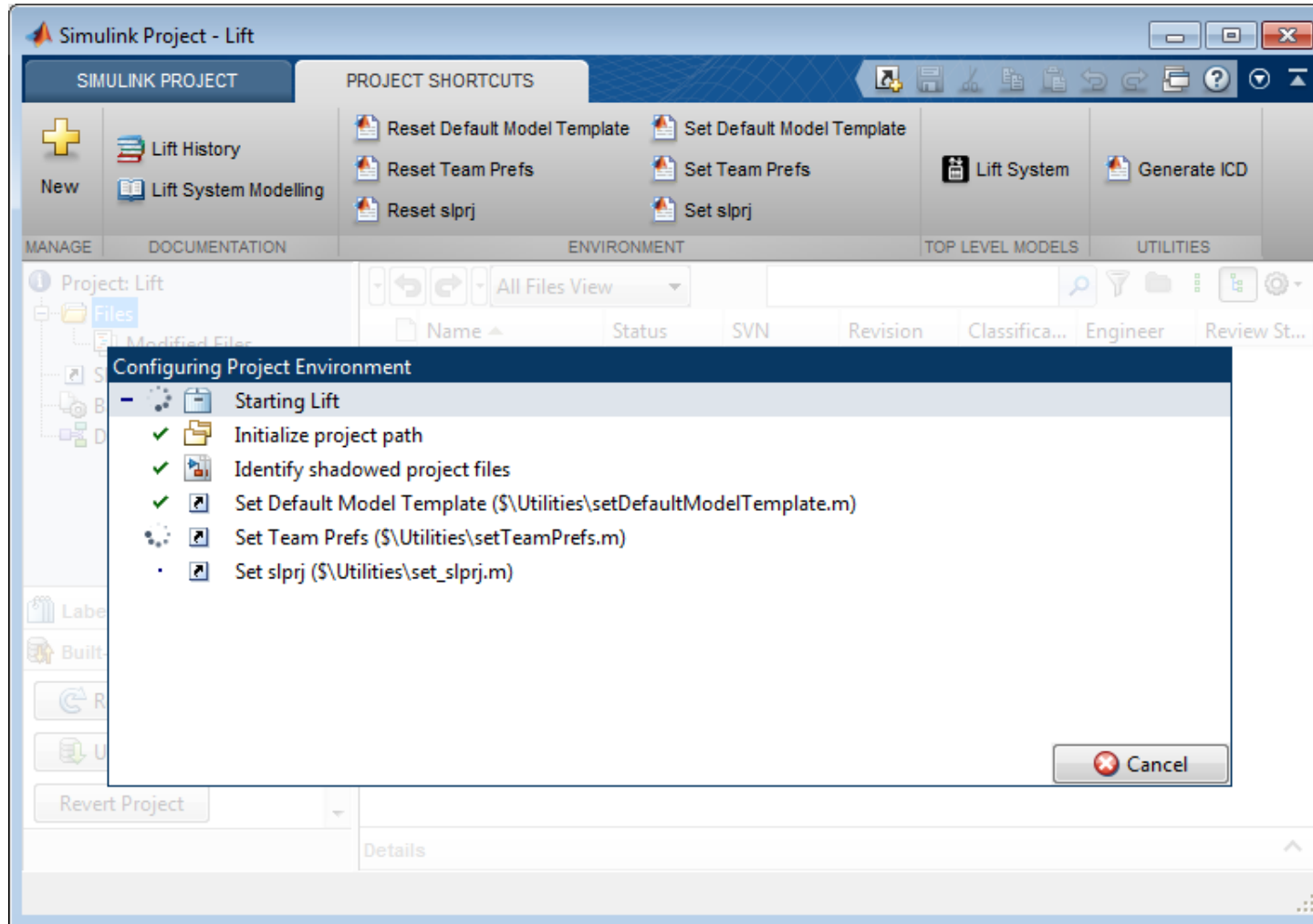
Details



Dependency Analysis – Modular Development

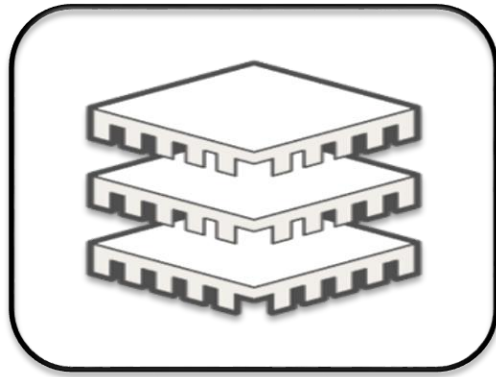
The screenshot displays the MATLAB Dependency Analysis tool interface. The main window shows a dependency graph for a project named 'lift_system_orig'. The graph consists of several nodes representing files, with arrows indicating dependencies. The nodes include 'lift_system.sbx', 'generic_motor.slx', 'lift_door_controller.slx', 'lift_inertia.slx', 'lift_inertia_pm.slx', 'lift_inertia_utils.slx', 'lift_door.req', and 'lift_system_Harness1.slx'. A callout box labeled 'Show model structure' points to the graph. Another callout box labeled 'List products required' points to a list of products on the right side of the interface. A third callout box labeled 'Highlight issues' points to a warning icon in the dependency graph. The right side of the interface shows a list of products required for the project, including MATLAB 9.1, Simulink 8.8, Simscape 4.1, Simscape Multibody 4.9, and Simulink Coder 8.11. Below this list, there is a section for 'Problem Files (1)' which includes 'LiftSystemModelling.docx'.

Task Automation – Configuring Project Environment



- Robustly configure the team environment
- For everyone
- Automatically

Simulink Addressing Scalability Challenges

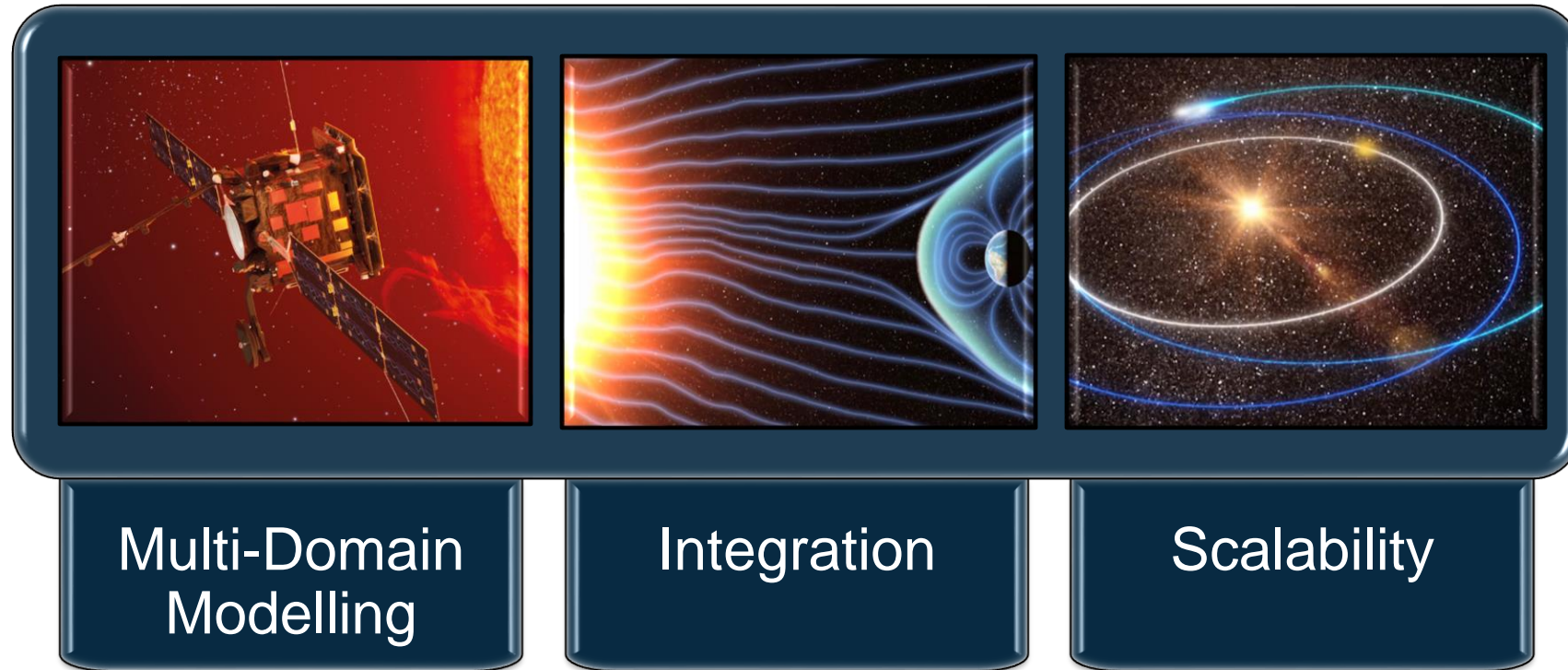


Performance



Team
Workflows

Enterprise Simulation Platform Enablers



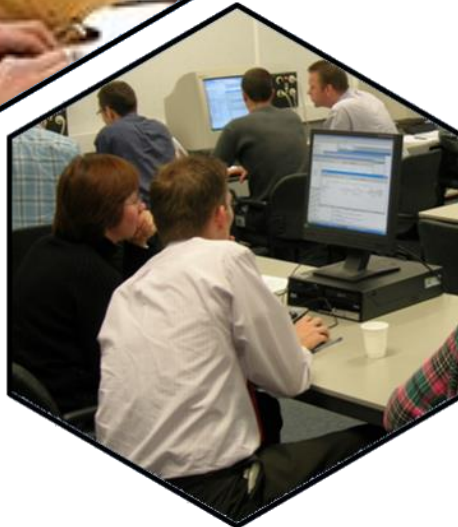
"No other tool gives us the multidomain simulation capability and block diagram environment in a way that is scalable to represent complex systems. That is why we use Simulink."

Andrew Pollard
Tessella

How to get started?

Public

On-Site



- Model-Based Design
- Stateflow
- Simscape

MATLAB[®]

Data Analytics

Data Processing and Visualization
 Statistics
 Machine Learning
 Optimization Techniques
 Parallel Computing

Application Development

Programming Techniques
 Building Interactive Applications
 Object-Oriented Programming

Code Generation

MATLAB Coder
 Interfacing with C-code

Application-Specific

Control System Design
 Signal Processing
 Communication Systems
 LTE Systems

Computational Finance

Risk Management
 Time-Series Modelling

Signal Processing

Using MATLAB
 Using Simulink

Image and Video Processing

Image Processing
 Computer Vision

SIMULINK[®]

Model-Based Design

Implementing MBD Workflow
 Model Management and Architecture
 Verification and Validation

Stateflow[®]

Event-Based Modeling

Simscape[™]

General Simscape[™]
 Simscape Multibody[™]
 Simscape Driveline[™]
 Simscape Fluids[™]
 Simscape Power Systems[™]

Code Generation

Rapid Prototyping and HIL-Simulation
 Embedded Systems
 FPGA Design
 Generating HDL Code
 Xilinx Zynq SoCs
 AUTOSAR

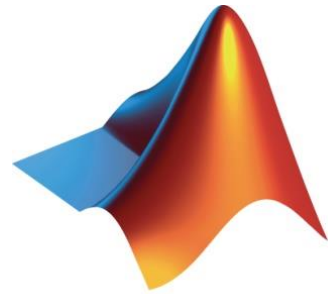
Code Integration

Integrating C and MATLAB

Polyspace[®]

Polyspace Code Prover[™]

<https://nl.mathworks.com/services/training.html>



MathWorks®

Accelerating the pace of engineering and science

© 2017 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.