



# Requirement Modeling

of Distributed Automotive Control Systems

*Presented by:*

***Nate Rolfes***

***Ford Motor Company***

*Material prepared for the Mathworks Automotive Conference, May 12, 2016*



# Pro Trailer Backup Assist

## Distributed Control System Overview

### Powertrain System

- Throttle Speed Controller for Speed Limiting
- Gear Shift Lever for State Logic

### Steering System

- Steering Torque Sensing
- Steering Controller Actuation and Logic

### Brake System

- Wheel Speed Sensors for Odometry
- Accelerometers for Vehicle Dynamics State Estimation
- Braking Controller Actuation for Speed Limiting



# Pro Trailer Backup Assist

## Distributed Control System Overview



### Camera System

- Rearview Camera for Trailer Angle Detection

# Pro Trailer Backup Assist

## Distributed Control System Overview

### Camera System

- Rearview Camera for Trailer Angle Detection
- Lighting for Night Usage

# Pro Trailer Backup Assist

## Distributed Control System Overview

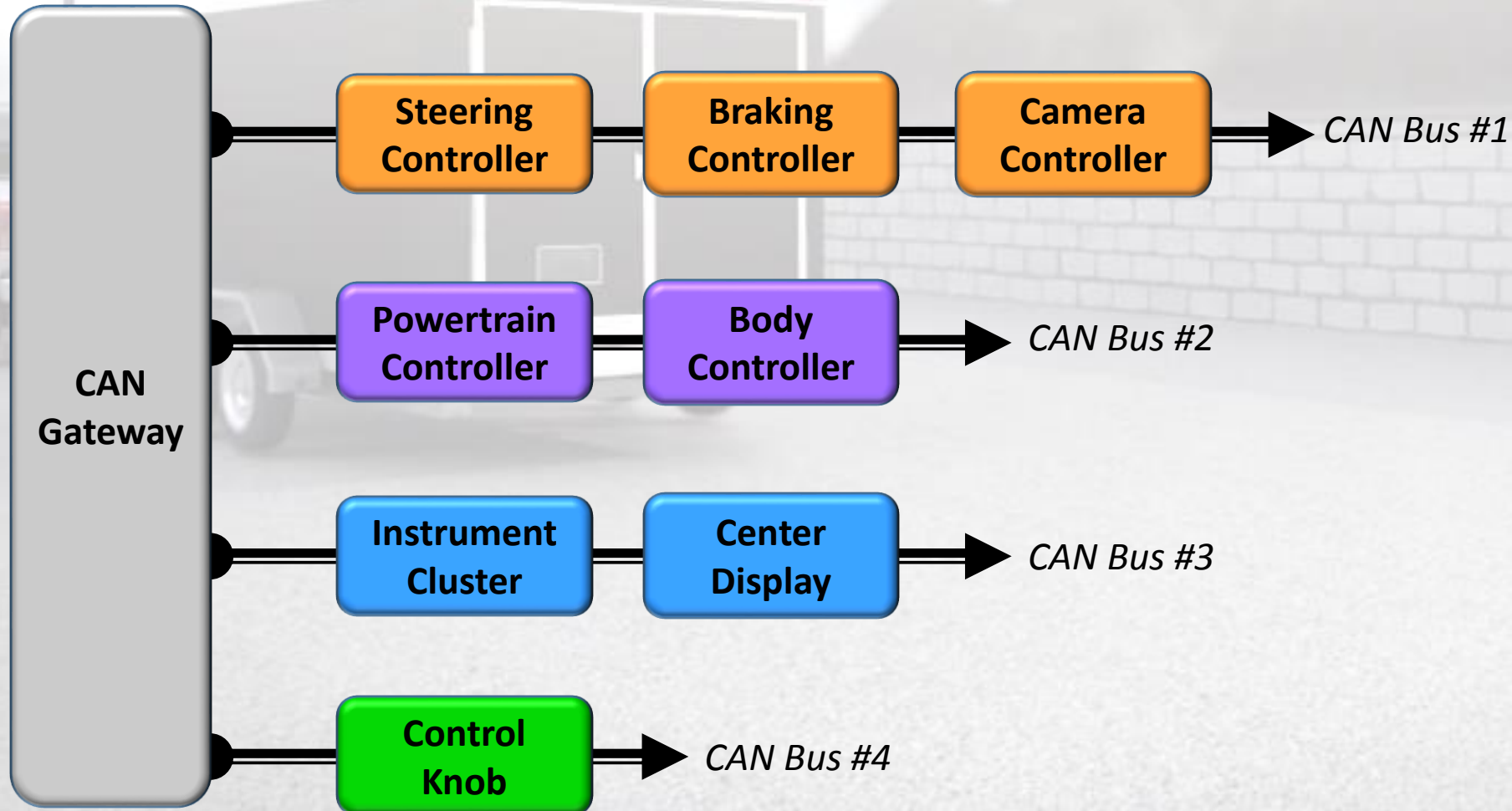
### HMI System: Driver Inputs

- Activation Switch and Control Knob
- Five-Way Buttons for Driver Inputs
- Cluster Display for Menu Selection and Instructions
- Center Console Display for Trailer View & Warnings



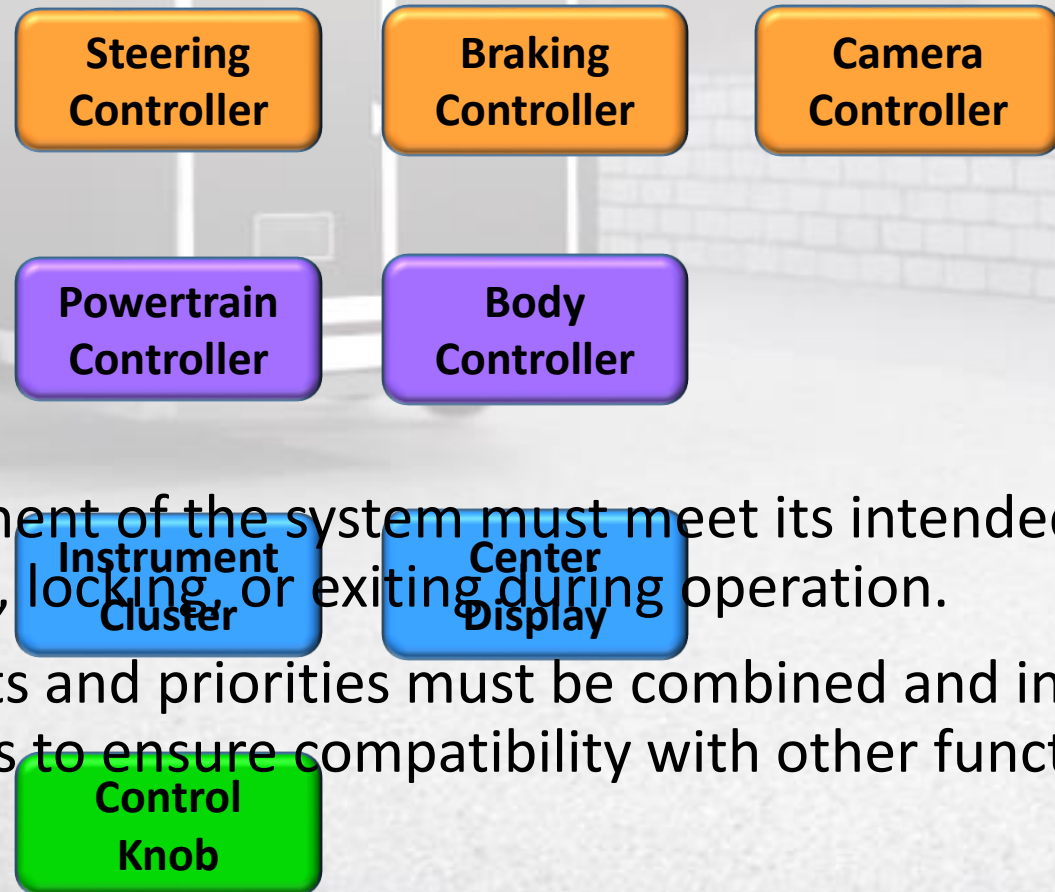
# Challenges of Pro Trailer Backup Assist

- There is not a standalone “Pro Trailer Backup Assist” Module
- The feature is a Distributed Logic Control System containing Eight ECU’s on four CAN buses connected through a CAN Gateway



# Challenges of Pro Trailer Backup Assist

- The control logic is designed based on engineering considerations, e.g.
  - Optimizing and sharing new functionality
  - Leveraging and adapting carryover functionality
  - Minimizing communication bandwidth



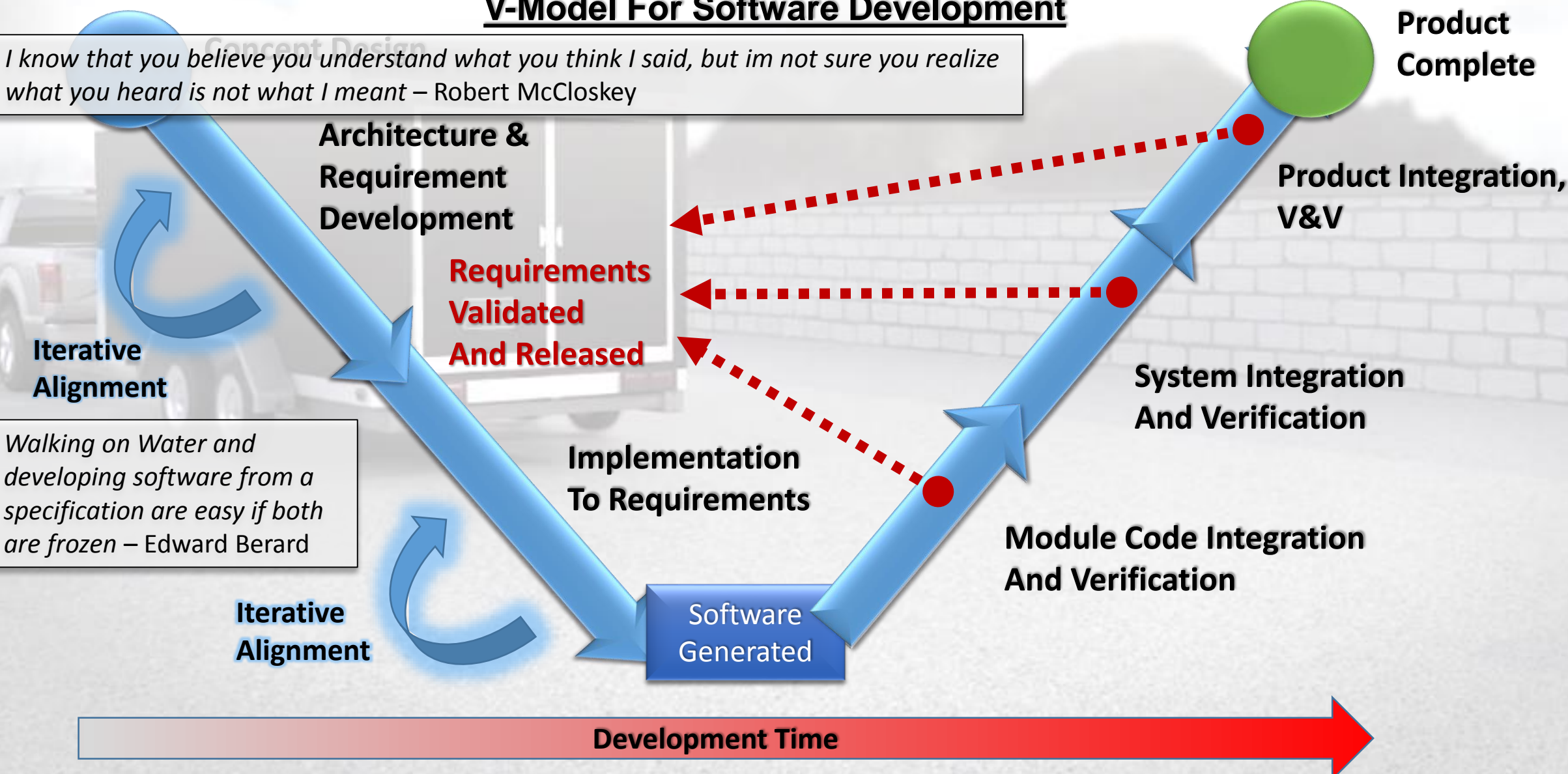
- Each component of the system must meet its intended **logical requirements** to prevent logic looping, locking, or exiting during operation.
- Requirements and priorities must be combined and integrated with existing functional requirements to ensure compatibility with other functional systems and interfaces.



# Challenges of Pro Trailer Backup Assist

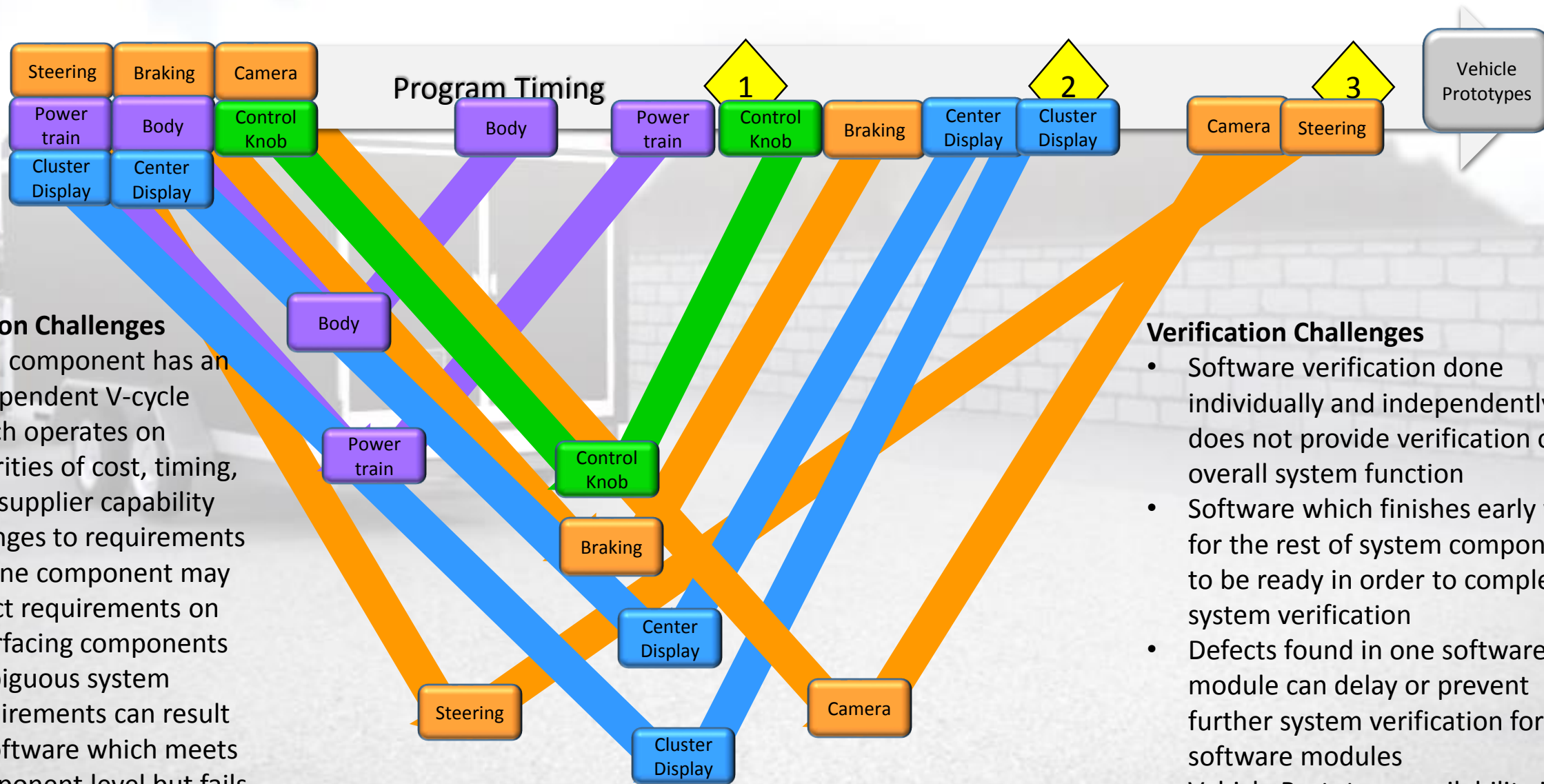
## V-Model For Software Development

*I know that you believe you understand what you think I said, but im not sure you realize what you heard is not what I meant – Robert McCloskey*



*Walking on Water and developing software from a specification are easy if both are frozen – Edward Berard*

# Challenges of Pro Trailer Backup Assist



## Validation Challenges

- Each component has an independent V-cycle which operates on priorities of cost, timing, and supplier capability
- Changes to requirements on one component may affect requirements on interfacing components
- Ambiguous system requirements can result in software which meets component level but fails the system level

## Verification Challenges

- Software verification done individually and independently does not provide verification of the overall system function
- Software which finishes early waits for the rest of system components to be ready in order to complete system verification
- Defects found in one software module can delay or prevent further system verification for other software modules
- Vehicle Prototype availability is too late to resolve critical defects

# Solutions of Pro Trailer Backup Assist

## 1. REQUIREMENT MODELING:

- A modeling methodology for Requirements which captures and simulates the logical parts to ensure the distributed control logical design of requirements works as intended prior to release for software implementation

## 2. DISTRIBUTED NETWORK SIMULATION:

- A simulation environment which can link multiple Controller modules, CAN Networks, Driver and Vehicle Interactions.
- It can simulate both MIL (Virtual) and HIL (Hardware) in real-time and each controller can be switched in real-time to either the MIL or HIL version. It can test all systems together or target systems individually at the system engineer's discretion

## 3. VALIDATION AND VERIFICATION TOOL:

- A tool that can work effectively throughout the Software V process to:
  - ✓ Test and validate requirement models (Down the System V)
  - ✓ Verify that software components and module outputs match the requirement model behavior (Up the System V)

# What is Requirement Modeling?

**TIME**

## Remembering the Apollo 11 Moon Landing With the Woman Who Made It Happen

Lily Rothman @lilyrothman | July 20, 2015



“...Part of what had made Hamilton’s work so effective was that she tested everything so rigorously, **in a simulator that could demonstrate the “system of systems” at work,** and the relationship between the software, the hardware and the astronaut. “We couldn’t run something up to the moon,” she says. But they could run lots of tests on the ground. **Hamilton’s team found that nearly three-quarters of them were interface errors, like conflicts in timing or priority...**”



Margaret Hamilton

<http://time.com/3948364/moon-landing-apollo-11-margaret-hamilton/>

# What is Requirement Modeling?

2013-01-2237

## Requirements Modeling and Automated Requirements-Based Test Generation

John Lee and Jon Friedman  
MathWorks

Copyright © 2014 SAE International

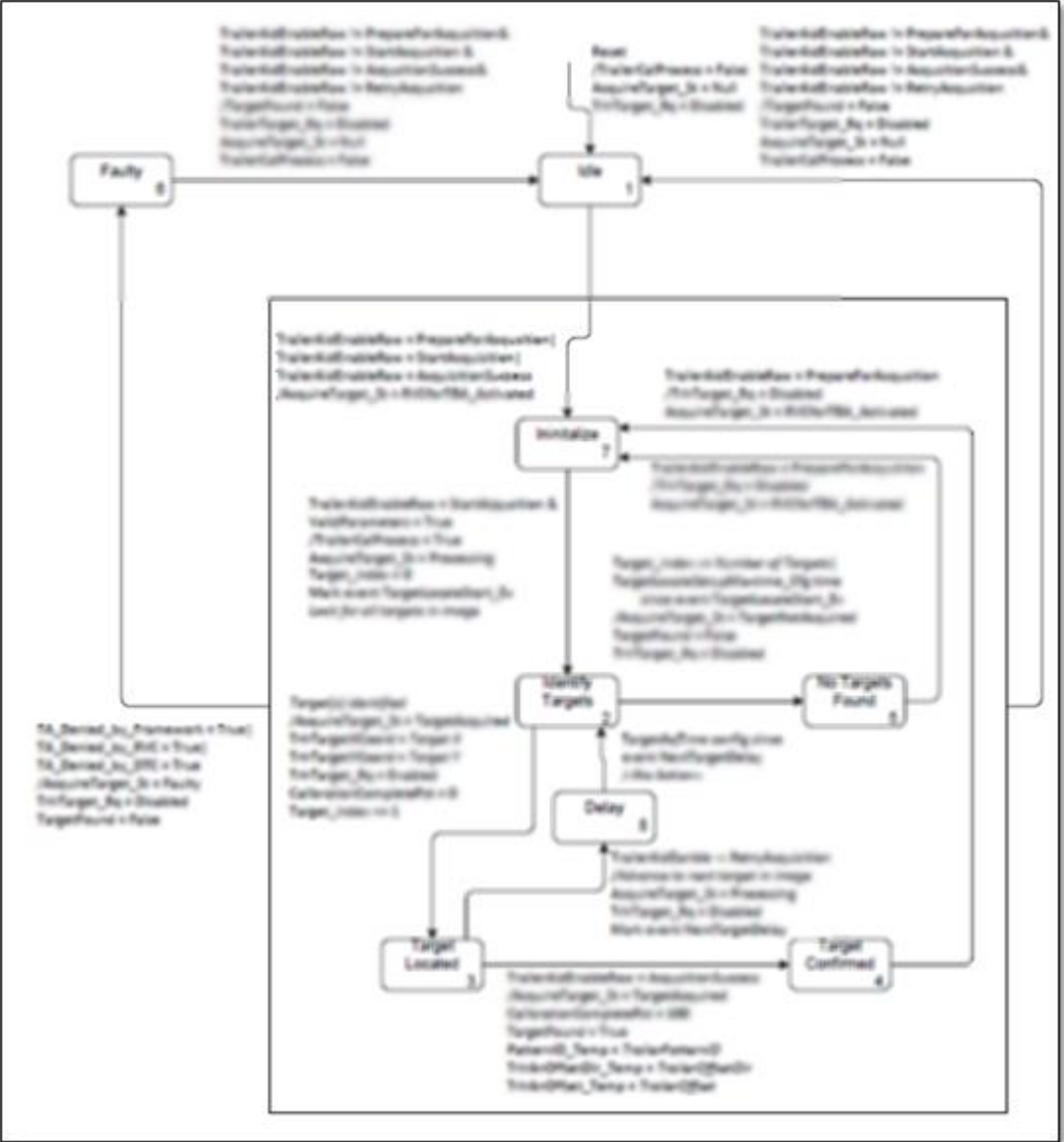
“...The goal of requirements models is **to capture the functional requirement in a clear, concise, analyzable and executable manner, which is typically not possible with natural language.**

The requirements models can then be used to **evaluate the interaction and compatibility of requirements from disparate sources as well as to develop tests and acceptance criteria (or expected outputs).**

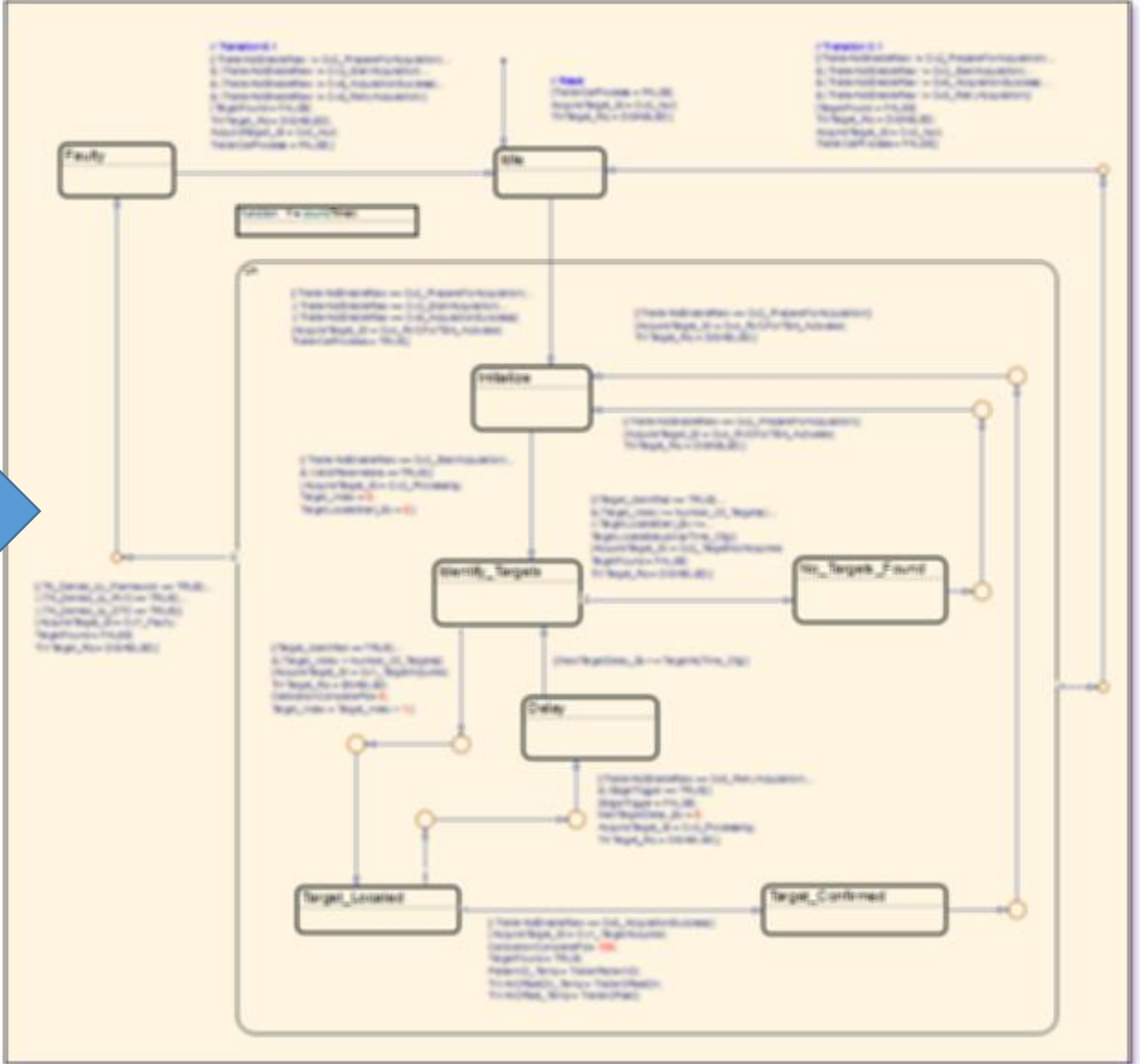
The use of the requirements models for test creation enables engineers to assess the completeness of the tests **using different notions of coverage on the requirements model...**”

# Requirement Modeling Example

Paper Specification State Machine:



Stateflow Requirement Model:





# Solutions of Pro Trailer Backup Assist

## 1. REQUIREMENT MODELING:

- A modeling methodology for Requirements which captures and simulates the logical parts to ensure the distributed control logical design of requirements works as intended prior to release for software implementation

## 2. DISTRIBUTED NETWORK SIMULATION:

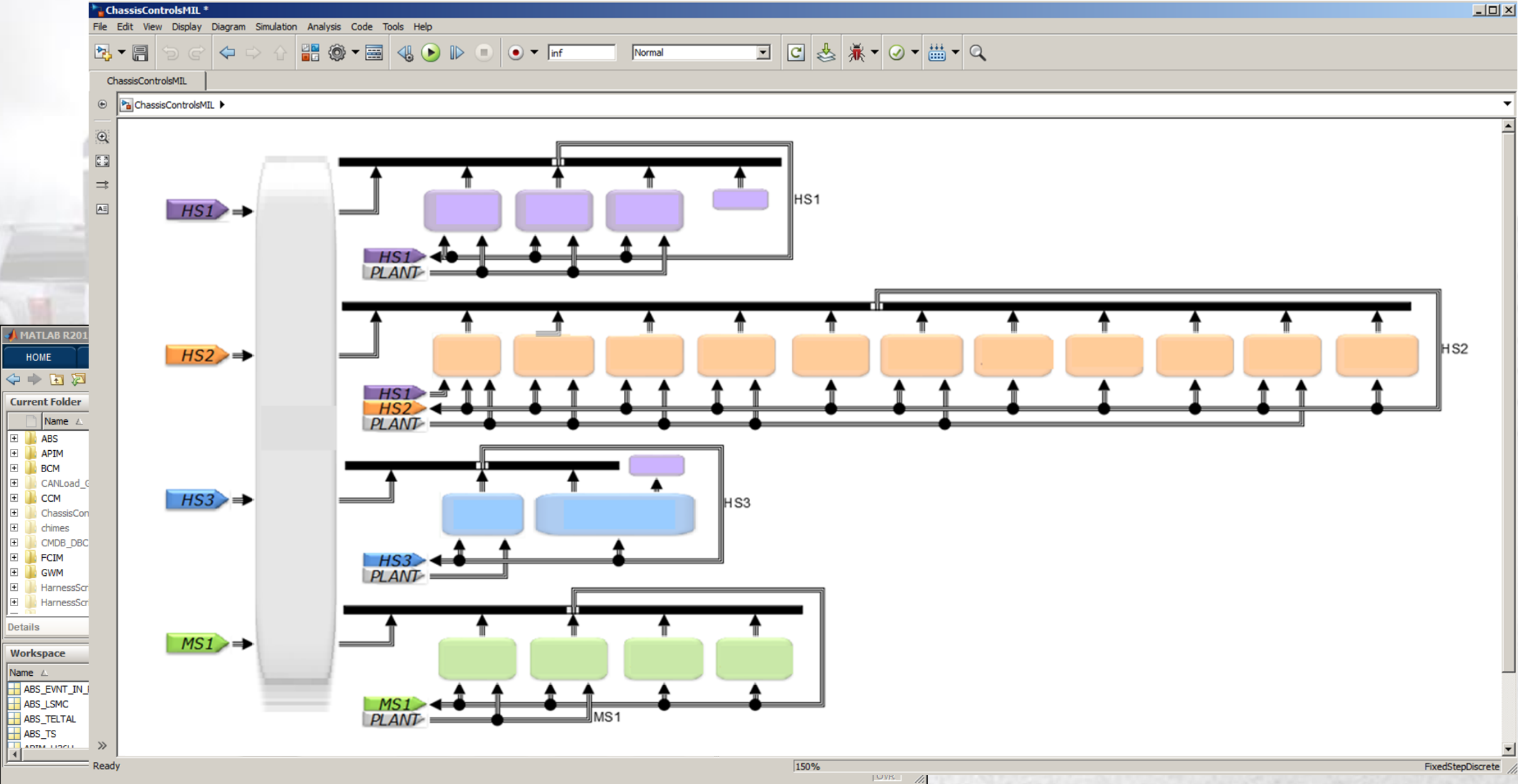
- A simulation environment which can link multiple Controller modules, CAN Networks, Driver and Vehicle Interactions.
- It can simulate both MIL (Virtual) and HIL (Hardware) in real-time and each controller can be switched in real-time to either the MIL or HIL version. It can test all systems together or target systems individually at the system engineer's discretion

## 3. VALIDATION AND VERIFICATION TOOL:

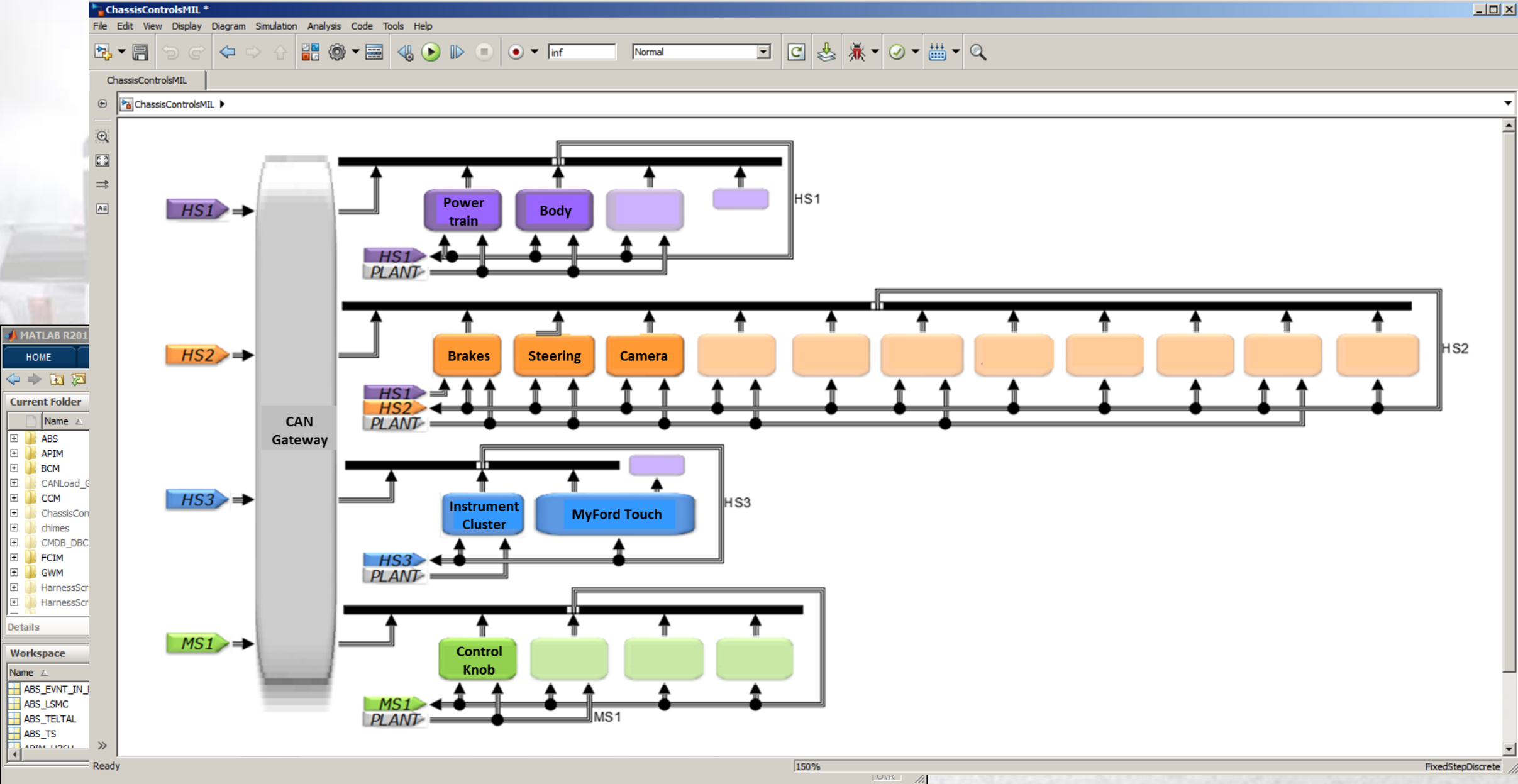
- A tool that can work effectively throughout the Software V process to:
  - ✓ Test and validate requirement models (Down the System V)
  - ✓ Verify that software components and module outputs match the requirement model behavior (Up the System V)



# Distributed Network Simulation MIL



# Distributed Network Simulation MIL



# Distributed Network Simulation MIL

The screenshot displays the ChassisControlsMIL simulation environment. At the top, a menu bar includes File, Edit, View, Display, Diagram, Simulation, Analysis, Code, Tools, and Help. Below the menu is a toolbar with various icons for file operations, simulation control, and search. The main workspace shows a vehicle model on a coordinate plane with axes ranging from -4 to 8. A large red text overlay reads "Simulator of Systems".

On the left, a MATLAB R2011b interface is visible, showing the current folder and workspace. The workspace contains files such as ABS\_EVTN\_IN\_1, ABS\_LSMC, ABS\_TELTAL, ABS\_TS, and ABS\_TSCU.

The central and right portions of the interface are dominated by the simulator GUI. The "simulator\_gui" window features a "Menu" with options: Display Mode, Trip/Fuel, Towing, Off Road, and Settings. Below the menu are controls for a 5-Way Control (directional pad with OK) and Trailer Backup Assist (TBA OFF). The "Trailer Type Connected" dropdown is set to "None".

The "ChassisControlsMIL\_GUI" window provides detailed simulation controls and data. It includes "Simulation Controls" with buttons for START, PAUSE, STOP, CAN BUS, and GPS. A speed readout shows 705.2. Other controls include MEM CHECK, RESET MODEL, SIM Parameters, Helper, Edit Variants, Set Variants, and Edit GPS. The "Automatic Trans" dropdown is set to GUI. Vehicle parameters include P552 3584 and P285/70R17 2522. The "Vehicle State" section shows gear (P), clutch (0), brake (N), steering (0), and suspension (0) status, along with a "CLOSED" door status. The "Driver Controls" section includes steering wheel input, brake, accelerator, and speed limits (25 Fwd, 15 Rwd). The "5-Way Control" window is also visible at the bottom right.

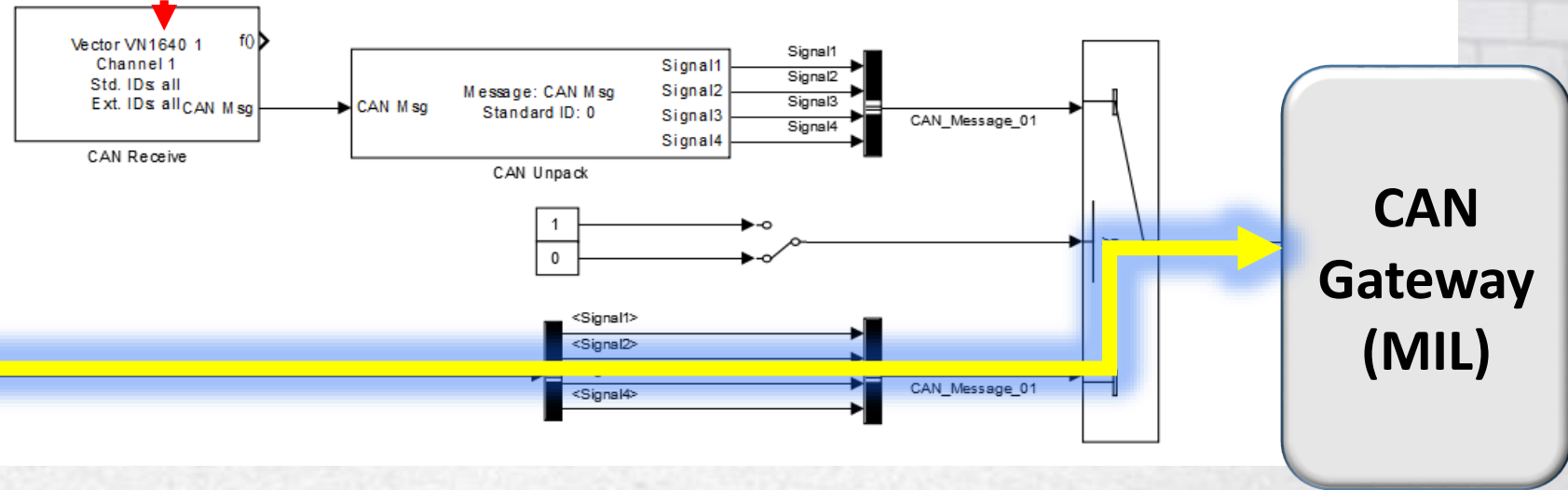
At the bottom of the MATLAB window, the status bar shows "Ready", "150%", and "FixedStepDiscrete".

# Adding Hardware using Vehicle Network Toolbox

Instrument Cluster  
Hardware (HIL)



Vehicle Network Toolbox




Instrument Cluster  
Requirement Model  
(MIL)

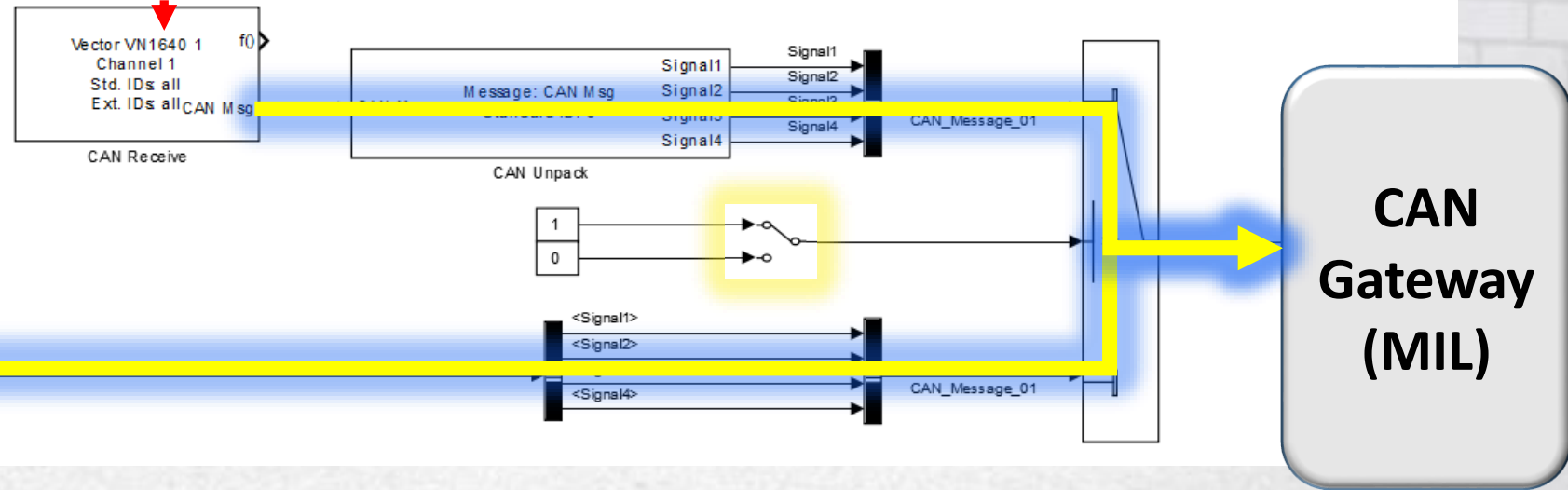
CAN  
Gateway  
(MIL)

# Adding Hardware using Vehicle Network Toolbox

Instrument Cluster  
Hardware (HIL)



 Vehicle Network Toolbox



Instrument Cluster  
Requirement Model  
(MIL)

# Distributed Network Simulation MIL & HIL

The screenshot displays the ChassisControlsMIL simulation environment. The main window shows a distributed network architecture centered around a CAN Gateway. The network is divided into four hierarchical levels:

- HS1 (High-Speed 1):** Contains Power train, Body, and other high-level control blocks.
- HS2 (High-Speed 2):** Contains Brakes, Steering, and other chassis control blocks.
- MS1 (Medium-Speed 1):** Contains a TRAILER BACK UP control unit and other medium-speed control blocks.
- PLANT:** Includes a digital instrument cluster, a camera view of the vehicle, and a physical chassis model.

External nodes are connected to the CAN Gateway via HS1, HS2, and MS1 buses. The interface includes a MATLAB R2019a workspace on the left, a simulation control panel at the bottom right, and a vehicle data display at the bottom center.

**Simulation Control Panel (simulator\_gui):**

- Buttons: START, PAUSE, STOP, CAN BUS, GPS
- Simulation Speed: 705.2
- Menu: Display Mode, Trip/Fuel, Towing, Off Road, Settings
- 5-Way Control: OK, Cancel, Up, Down, Left/Right
- Trailer Backup Assist: TBA OFF
- Trailer Type Connected: None

**Vehicle State Panel (ChassisControlsMIL\_GUI):**

- Automatic Trans: GUI
- Vehicle State: 0 P N -0 0 0 CLOSED
- Vehicle Speed: 0 kph
- Vehicle Yaw Rate: 0 deg/sec
- Vehicle Position: X 0 m, Y 0 m, Angle 0 deg
- Steering Wheel Angle: 0 deg
- Steering Wheel Input: 0 Nm
- Brake: 0 Nm
- Accelerator: 0 %
- Ignition Status: Run
- AT Gear: P R N D L
- Trailer Attach: 0
- Door/Res: Factory/Res

# Distributed Network Simulation MIL & HIL

The image displays a complex simulation environment for vehicle chassis controls. The main window, titled "ChassisControlsMIL", shows a network diagram with a central "CAN Gateway" connected to four hierarchical simulation levels: HS1 (Power train, Body), HS2 (Brakes, Steering, Cam), HS3 (Instrument cluster), and MS1 (Trailer Backup Assist). Each level is connected to a corresponding hardware-in-the-loop (HIL) plant. A physical vehicle chassis is shown in the center, with various components like the engine, transmission, and wheels connected to the simulation blocks.

On the left, a "MATLAB R201" window shows a file explorer with folders like ABS, APIM, BCM, and CANLoad\_C, and a workspace with variables like ABS\_EVNT\_IN\_I and ABS\_LSMC.

In the bottom right, a "simulator\_gui" window displays the "Pro Trailer Backup Assist" interface. It features a central display with the text "Backup Slowly Turn Knob to Steer" and "Press Knob to Exit". Below the display are controls for gear selection (P, R, N, D, L, M), ignition status (Run), vehicle speed (6 kph), and steering wheel angle. A "Trailer Backup Assist" knob is shown with "TBA ON" and "TBA OFF" buttons.

At the bottom right, a "ChassisControlsMIL\_GUI" window shows simulation controls including "START", "PAUSE", "STOP", "CAN BUS", and "GPS" buttons. It displays real-time data such as "726.9 MEM CHECK", "Automatic Trans GUI", and "Vehicle State" (6.8 R 0 0 0 CLOSED).

# Solutions

## 1. REQUIREMENT MODELING:

- A modeling methodology for Requirements which captures and simulates the logical parts to ensure the distributed control logical design of requirements works as intended prior to release for software implementation

## 2. DISTRIBUTED NETWORK SIMULATION:

- A simulation environment which can link multiple Controller modules, CAN Networks, Driver and Vehicle Interactions.
- It can simulate both MIL (Virtual) and HIL (Hardware) in real-time and each controller can be switched in real-time to either the MIL or HIL version. It can test all systems together or target systems individually at the system engineer's discretion

## 3. VALIDATION AND VERIFICATION TOOL:

- A tool that can work effectively throughout the Software V process to:
  - ✓ Test and validate requirement models (Down the System V)
  - ✓ Verify that software components and module outputs match the requirement model behavior (Up the System V)





# Validation and Verification Tool

- Based on previous experience and not-so-successful attempts with existing verification tools, I developed a unique verification tool that would integrate seamlessly into the Distributed Network Simulation environment.

## STEP 1: Define the Test Case

### Simple Trailer Backup Assist Test Case

1. Driver activates Trailer Backup Assist (Press Button)
2. Driver begins to back-up trailer for a few seconds (Shift to Reverse, Accelerator Pedal)
3. Driver stops the vehicle (Depress Brake Pedal)
4. Driver deactivates Trailer Backup Assist (Press Button)

# Validation and Verification Tool

## STEP 2: Simulate and Record the Test Case

The screenshot displays the ChassisControlsMIL GUI interface, which is used for simulating and recording test cases. The interface is divided into several main sections:

- Simulation Controls:** Includes buttons for START (green), PAUSE (yellow), STOP (red), CAN BUS (purple), and GPS (red). It also shows a speedometer reading of 440.4 and buttons for MEM CHECK, RESET MODEL, and Edit GUI. Other controls include SIM Parameters, Helper, Edit Variants, Set Variants, and Edit GPS.
- Vehicle State:** Displays gear selection (0, P, N, 0, 0, 0, CLOSED) and various vehicle parameters such as VehSpd, GearAt, GearMt, SvaComp, SvaOS, Swl, Eac, YawAng, YawRt, HitchAng, WhlDir, and RqSwa. It also shows DM8 Disp, Cl, IPC L1, Simulator, and Batt (Norm (13.8V)).
- Driver Controls:** Features a steering wheel input (HWT Nm), Brake, Accel, and Speed Limits (25, 15). It includes buttons for Release (1.5), Hold, Sine (0.2), AT Gear (P, R, N, D, L), TrailerAttach, Ignition\_Status (Off, Acc, Strt, Run), Turn Signal (Left, Off, Right), and DoorAjar (FactoryRes).
- Menu:** Offers options for Display Mode (Trip/Fuel, Towing, Off Road, Settings) and a 5-Way Control panel with directional arrows and an OK button. It also includes Trailer Backup Assist (TBA OFF) and Trailer Type Connected (None).
- Simulation Parameters:** Shows Automatic Trans (GUI), P552 3584, and P265/70R17 2522.
- Replay Section:** Includes buttons for Select/Load replay, Reset replay, and Start/Stop replay. It also displays Replay info (file/sample time) and Replay status (0, 0, 0).
- Logging File Conversion:** Features buttons for Select/Load log file, Generate playback, and Generate output file. It also includes a checkbox for Convert first output only.
- Harness Configuration:** Includes a Select config file button and a Current configuration file field (mappingFile.xlsx).
- Harness and Logging:** Includes a Create harness button, a ChassisControlsMIL dropdown, and a Current log file field (TestCase7).
- Logging File Conversion:** Includes a Log file to convert field (TestCase6) and a checkbox for Convert first output only.

The interface is overlaid on a MATLAB R201x workspace, showing a file explorer on the left and a workspace area at the bottom. The status bar at the bottom indicates 'Ready', '150%', and 'FixedStepDiscrete'.

# Validation and Verification Tool

## STEP 3: Generate Test Case Replay Script and Master Report

**TEST CASE SCRIPT**

Time	Driver Input	HMI Request	HMI Status	Camera Status	Setup Status	Steering Angle	Vehicle Speed
0		1 HMI	Inactive	Null	Inactive	0 deg	0 kph
2.82	TBA Button Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
3.14	TBA Button Not Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
4.26	Down Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
4.5	Down Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.54	Up Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.78	Up Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
6.86	Ok Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
7.06	Ok Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
16.94	Shift gear to Reverse	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
18.78	Accel pedal 25 %	2 HMI	ActivateTba	Null	Inactive	0 deg	0 kph
19.14	Accel pedal 50 %	2 HMI	ActivateTba	TbaActive	Inactive	0 deg	0 kph
19.66	Accel pedal 75 %	2 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
20.98	Accel pedal 50 %	4 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
21.3	Accel pedal 25 %	4 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
21.72	Accel pedal 0 %	14 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
22.34	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
22.76	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
23.16	Brake Pedal 75 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
24.12	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.004036 kph
24.46	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	1.1564 kph
24.82	Brake Pedal 0 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	6.101 kph
26.64	TBA Button Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.9395 kph
26.9	TBA Button Not Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.7276 kph
30.1	Ok Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.4148 kph
30.42	Ok Not Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	3.9163 kph
32.04	Shift gear to Park	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.98992 kph
30.1	Ok Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
30.42	Ok Not Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
32.04	Shift gear to Park	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
38.86	Shift gear to Park	1 HMI	Inactive	Null	Inactive	0 deg	0 kph

**TEST CASE MASTER**

# Validation and Verification Tool

## STEP 4: Replay and Record the Test Case

The screenshot displays the ChassisControlsMIL GUI interface, which is used for simulation and control of a vehicle system. The interface is divided into several main sections:

- Simulation Controls:** Located at the top right, it includes buttons for START (green), PAUSE (yellow), STOP (red), CAN BUS (purple), and GPS (red). It also shows a speedometer reading of 587.6 and buttons for MEM CHECK, RESET MODEL, and Edit GUI. Below these are SIM Parameters, Helper, Edit Variants, Set Variants, and Edit GPS.
- Vehicle State:** This section displays various vehicle parameters in a grid format:

0	P	N	0	0	0	CLOSED
VehSpd	GearAt	GearMt	SwaComp	SwaOS	Swt	Eac
-0	0	0	TC		Unk	0
YawAng	YawRt	HitchAng			WhiDir	RqSwa
DM8 Disp	Cl	IPC L1	Simulator	Batt	Norm (13.8V)	
- Driver Controls:** This section includes controls for Steering Wheel Input (HWT Nm), Brake, Accel, and Speed Limits (25 Fwd, 15 Rwd). It also features a gear selector (P, R, N, D, L, M) and buttons for AT Gear, TrailerAttoh, DoorAjar, and FactoryRes.
- Vehicle State Table:** A table showing various vehicle parameters and their current values:

Ignition Status	Run	Vehicle Speed	0 kph	Brake	0 Nm	Accelerator	0 %
Vehicle Yaw Rate	...	Vehicle Position	X: 173 m, Y: 0 m, Angle: ...	Steering Wheel Angle	0 deg	0 Nm	
- 5-Way Control:** A central control panel with a 5-way directional pad and buttons for OK, TBA OFF, and Trailer Type Connected (None).
- Simulation Controls (Bottom Right):** A panel with buttons for APA, 360, TBA, and 5-Way Control.

The interface also includes a MATLAB R201x workspace on the left side, showing a file explorer and a list of variables in the workspace. The status bar at the bottom indicates the simulation is running at 150% zoom and is in a FixedStepDiscrete mode.

# Validation and Verification Tool

## STEP 5: Compare Test Case Results against Master

Time	Driver Input	HMI Request	HMI Status	Camera Status	Setup Status	Steering Angle	Vehicle Speed
0		1 HMI	Inactive	Null	Inactive	0 deg	0 kph
2.82	TBA Button Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
3.14	TBA Button Not Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
3.26		2 HMI	Inactive	Null	Inactive	0 deg	0 kph
4.26	Down Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
4.5	Down Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.54	Up Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.78	Up Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
6.86	Ok Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
7.06	Ok Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
7.24		2 HMI	ActivateTba	Null	Inactive	0 deg	0 kph
7.32		2 HMI	ActivateTba	TbaActive	Inactive	0 deg	0 kph
7.46		2 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
7.58		4 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
7.84		4 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
15.46		14 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
16.94	Shift gear to Reverse	14 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
17.5		5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
18.78	Accel pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
19.14	Accel pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.004036 kph
19.66	Accel pedal 75 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	1.1564 kph
20.98	Accel pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	6.101 kph
21.3	Accel pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.9395 kph
21.72	Accel pedal 0 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.7276 kph
22.34	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.4148 kph
22.76	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	3.9163 kph
23.16	Brake Pedal 75 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.98992 kph
24.12	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
24.46	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
24.82	Brake Pedal 0 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
26.64	TBA Button Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
26.9	TBA Button Not Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
27.02		13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
30.1	Ok Pressed	13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
30.42	Ok Not Pressed	13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
32.04		1 HMI	Inactive	Null	Inactive	0 deg	0 kph
38.86	Shift gear to Park	1 HMI	Inactive	Null	Inactive	0 deg	0 kph

Time	Driver Input	HMI Request	HMI Status	Camera Status	Setup Status	Steering Angle	Vehicle Speed
0		1 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.1	TBA Button Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.42	TBA Button Not Pressed	1 HMI	Inactive	Null	Inactive	0 deg	0 kph
5.54		2 HMI	Inactive	Null	Inactive	0 deg	0 kph
6.54	Down Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
6.78	Down Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
7.82	Up Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
8.06	Up Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
9.14	Ok Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
9.34	Ok Not Pressed	2 HMI	Inactive	Null	Inactive	0 deg	0 kph
9.72		2 HMI	ActivateTba	Null	Inactive	0 deg	0 kph
9.8		2 HMI	ActivateTba	TbaActive	Inactive	0 deg	0 kph
9.94		2 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
10.06		4 HMI	DeactivateTba	TbaActive	Inactive	0 deg	0 kph
10.32		4 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
17.94		14 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
19.22	Shift gear to Reverse	14 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
19.78		5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
21.06	Accel pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
21.42	Accel pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.004036 kph
21.94	Accel pedal 75 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	1.1564 kph
23.26	Accel pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	6.101 kph
23.58	Accel pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.9395 kph
24	Accel pedal 0 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.7276 kph
24.62	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	5.4148 kph
25.04	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	3.9163 kph
25.44	Brake Pedal 75 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0.98992 kph
26.4	Brake Pedal 50 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
26.74	Brake Pedal 25 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
27.1	Brake Pedal 0 %	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
28.92	TBA Button Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
29.18	TBA Button Not Pressed	5 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
29.3		13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
32.38	Ok Pressed	13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
32.7	Ok Not Pressed	13 HMI	Inactive	TbaActive	Inactive	0 deg	0 kph
34.32		1 HMI	Inactive	Null	Inactive	0 deg	0 kph
41.14	Shift gear to Park	1 HMI	Inactive	Null	Inactive	0 deg	0 kph

TEST CASE MASTER

TEST CASE ITERATIVE

- ABS\_EVNT\_INV
- ABS\_LSMC
- ABS\_TELTAL
- ABS\_TS
- ARM\_H2G1

Ready
150%
FixedStepDiscrete

# Validation and Verification Tool

## STEP 5: Compare Test Case Results against Master

Time	Value	Component	State	Speed	Component	State	Speed
4	2.8200000	TBA Button	11	1	1 HMI	Inactive	Null
5	3.1400000	TBA Button	11	0	1 HMI	Inactive	Null
6	3.2600000				2 HMI	Inactive	Null
7	4.2600000	Down Pres	7	1	2 HMI	Inactive	Null
8	4.5	Down Not	7	0	2 HMI	Inactive	Null
9	5.5400000	Up Pressed	6	1	2 HMI	Inactive	Null
10	5.7800000	Up Not Pre	6	0	2 HMI	Inactive	Null
11	6.8600000	Ok Pressed	10	1	2 HMI	Inactive	Null
12	7.0600000	Ok Not Pre	10	0	2 HMI	Inactive	Null
13	7.2400000				2 HMI	ActivateTb	Null
14	7.3200000				2 HMI	ActivateTb	TbaActive
15	7.4600000				2 HMI	Deactivate	TbaActive
16	7.5800000				4 HMI	Deactivate	TbaActive
17	7.8400000				4 HMI	Inactive	TbaActive
18	15.4600000				14 HMI	Inactive	TbaActive
19	16.9400000	Shift gear t	4	1	14 HMI	Inactive	TbaActive
20	17.5				5 HMI	Inactive	TbaActive
21	18.7800000	Accel pedal	3	25	5 HMI	Inactive	TbaActive
22	19.1400000	Accel pedal	3	50	5 HMI	Inactive	TbaActive
23	19.6600000	Accel pedal	3	75	5 HMI	Inactive	TbaActive
24	20.9800000	Accel pedal	3	50	5 HMI	Inactive	TbaActive
25	21.3000000	Accel pedal	3	25	5 HMI	Inactive	TbaActive
26	21.7200000	Accel pedal	3	0	5 HMI	Inactive	TbaActive
27	22.3400000	Brake Peda	2	25	5 HMI	Inactive	TbaActive
28	22.7600000	Brake Peda	2	50	5 HMI	Inactive	TbaActive
29	23.1600000	Brake Peda	2	75	5 HMI	Inactive	TbaActive
30	24.1200000	Brake Peda	2	50	5 HMI	Inactive	TbaActive
31	24.4600000	Brake Peda	2	25	5 HMI	Inactive	TbaActive
32	24.8200000	Brake Peda	2	0	5 HMI	Inactive	TbaActive
33	26.6400000	TBA Button	11	1	5 HMI	Inactive	TbaActive
34	26.9000000	TBA Button	11	0	5 HMI	Inactive	TbaActive
35	27.0200000				13 HMI	Inactive	TbaActive
36	30.1000000	Ok Pressed	10	1	13 HMI	Inactive	TbaActive
37	30.4200000	Ok Not Pre	10	0	13 HMI	Inactive	TbaActive
38	32.0400000				1 HMI	Inactive	Null
39	38.8600000	Shift gear t	4	0	1 HMI	Inactive	Null

- ABS\_EVTN\_IN
- ABS\_LSMC
- ABS\_TELTAL
- ABS\_TS
- ARM\_H2CU

Ready

150%

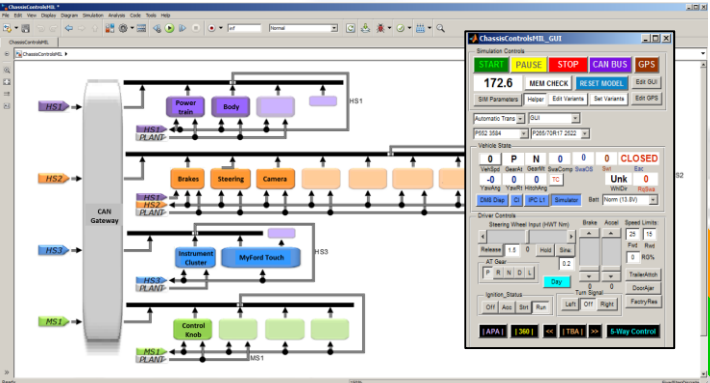
FixedStepDiscrete

# Validation and Verification Tool

- Recording a Master Test Case and generating a Master Test Report creates a document which captures the system outputs based on the Requirement Models.
- The Master Test Case behavior can be replayed repeatedly to verify the system for new software releases of each module for any MIL/HIL configuration of the Distributed Network Simulation.
- Master Test Reports can be provided to engineers and suppliers to define how their module should react in the system. They can be customized and targeted towards specific modules so that only the relevant test data is generated in the report.
- Iterative Test Reports can be compared against the Master to exactly identify logical defects within the context of Simulation Time and Driver Actions.
- Test Reports can be configured to include any system inputs, outputs, or parameters that exist in the simulation environment
- Can be used in conjunction with Coverage Tool to track coverage metrics.
- In conjunction with Vector CANape (CAN and Video logging), all test case data can be logged into a single synchronous timeline for evidence and review.

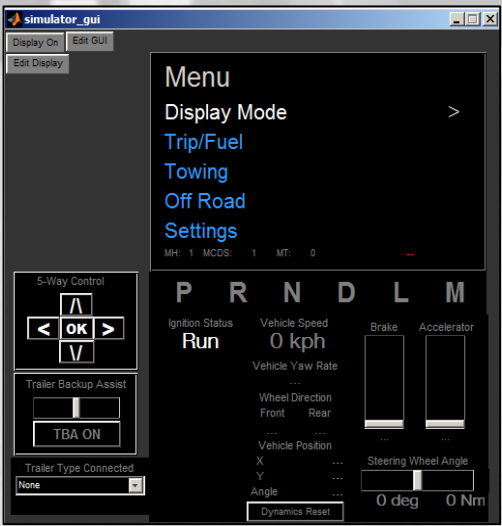
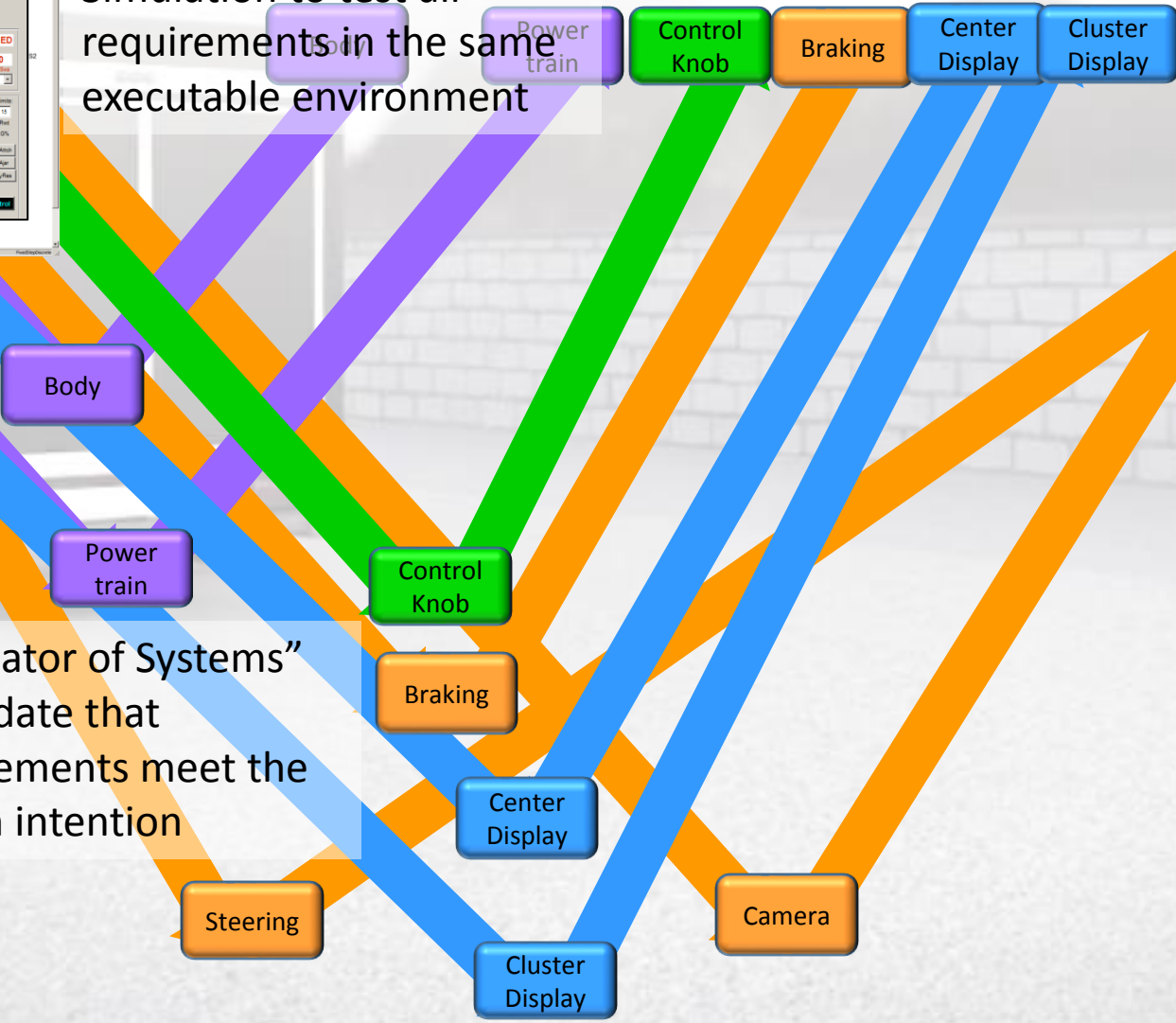


# Solutions Overview

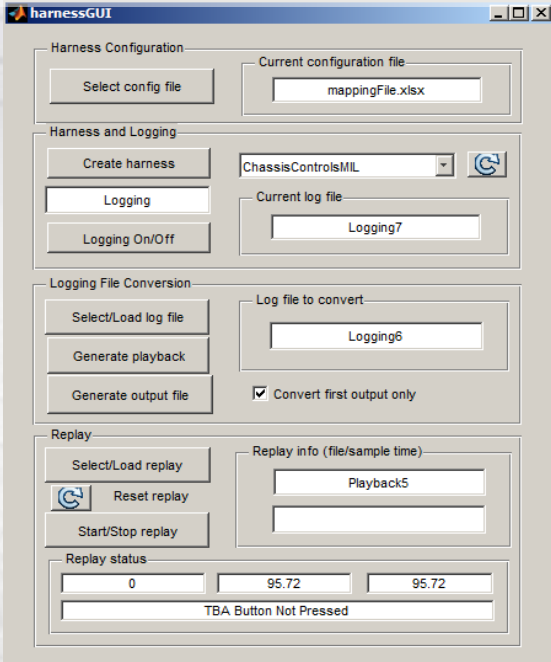


Requirement Model  
Simulation to test all requirements in the same executable environment

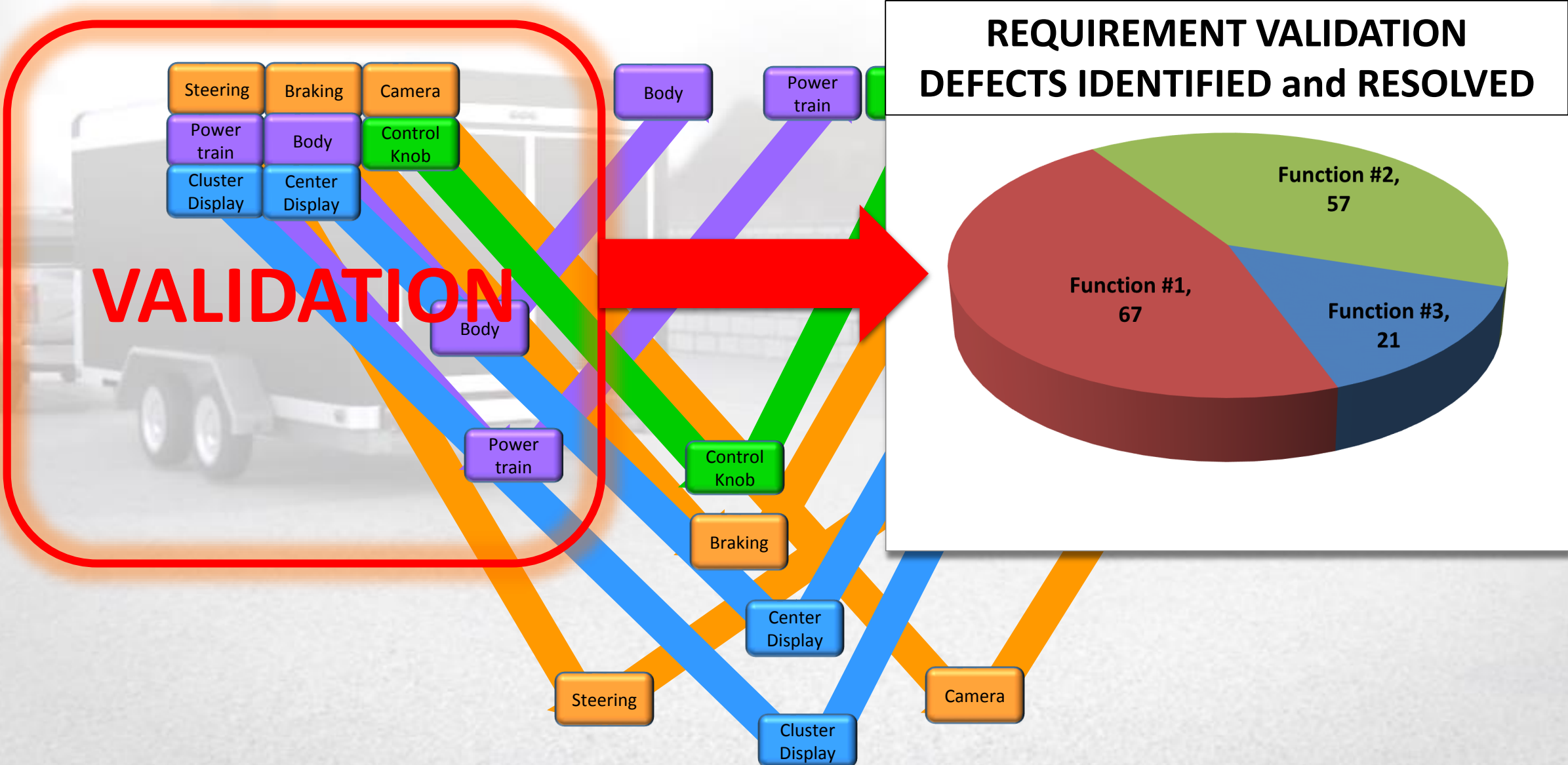
Validation and Verification  
Tool to record Master Test Cases and generate repeatable test scripts which can be replayed in any MIL/HIL integration configurations



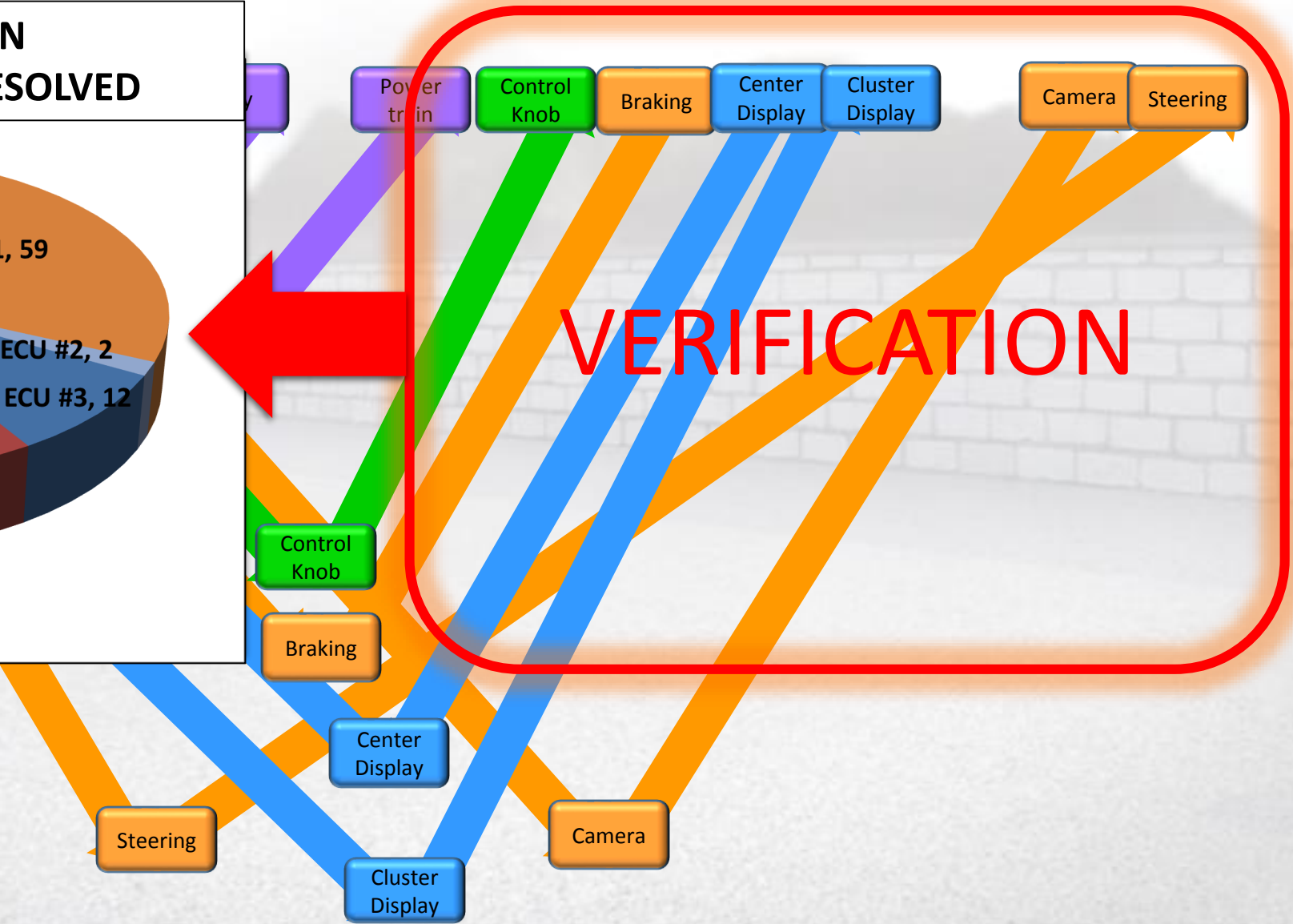
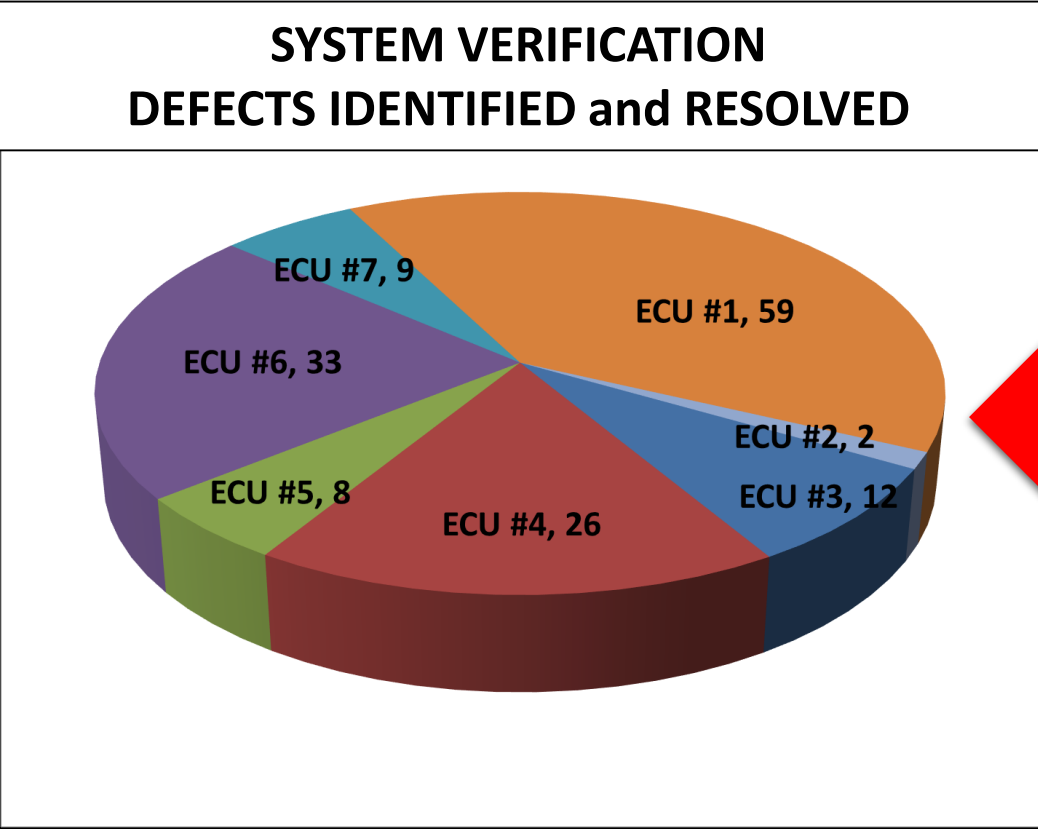
“Simulator of Systems”  
to validate that requirements meet the Design intention



# Results



# Results



# What's Next?

## Validation

*Concept → MIL*

Emphasis:

*Readability*

Output:

*Requirements*

## Verification

*SIL → HIL*

Emphasis:

*Testability, Traceability*

Output:

*Prototypes*

## Implementation

*MIL → SIL*

Emphasis:

*Efficiency, Compliance*

Output:

*Software*

# What's Next?

## Validation

*Concept → MIL*

Emphasis:

*Readability*

Output:

*Requirements*

- Requirement Validation step is often skipped, overlooked, or misunderstood.
- Requirement Validation skillsets and tools are undeveloped and unrecognized
- Few tools exist to simulate and validate requirements.
- An ideal tool would provide the ability to simulate and generate requirements from a model the same way that tools exist to generate, test, and verify code and hardware from a model.

# What's Next?

## Validation

*Concept* → *MIL*

Emphasis:

*Readability*

Output:

*Requirements*

- Requirement “modeling” is also done in formats that are non-executable.
- Translation from one tool, language, or format to another takes significant time and resource and introduces errors in translation.
- Requirement Modeling in Matlab is uniquely effective and efficient when code generation and verification is already done in Matlab – there is no translation needed!
- Building an executable model that can be used throughout the System V without translation is a **HUGE** efficiency gain and the essence of Model-Based Design.

**Thank you for your time and attention! 😊**

## **Acknowledgements**

**John Lee**

**Dr. Darrel Recker**

**Brad Hochrein**

**MathWorks**

**Ford Motor Company**

**Ford Motor Company**

**Comments and Feedback can be directed to:**

**Nate Rolfes**

**[nrolfes@ford.com](mailto:nrolfes@ford.com)**