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
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

![Diagram of CAN bus connections](image)

- CAN Bus #1: Steering Controller → Braking Controller → Camera Controller
- CAN Bus #2: Powertrain Controller → Body Controller
- CAN Bus #3: Instrument Cluster → Center Display
- CAN Bus #4: Control Knob
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
  - Meeting program timing

- 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 I’m 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

Development Time

Software Generated

Requirements Validated And Released

Implementation To Requirements

Module Code Integration And Verification

System Integration And Verification

Product Integration, V&V

Product Complete

Architecture & Requirement Development

Iterative Alignment

Iterative Alignment
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
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?

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

Margaret Hamilton

“...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...”

http://time.com/3948364/moon-landing-apollo-11-margaret-hamilton/
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:
Requirement Modeling Example
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

“Simulator of Systems”
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)

CAN Gateway (MIL)
Distributed Network Simulation MIL & HIL
Distributed Network Simulation MIL & HIL
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

- Most verification tools are not intended to handle “simulation of systems” with driver-in-the-loop. Most are designed and focused for unit-level testing and verification.
- Simulink Signal Generator tool works for unit testing of simple models and test cases with a limited and predictable number of inputs and outputs.
- It becomes cumbersome to modify and maintain Signal Generator for highly complex or distributed logic control models with hundreds of potential inputs and outputs.
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
Validation and Verification Tool

STEP 3: Generate Test Case Replay Script and Master Report
Validation and Verification Tool

STEP 4: Replay and Record the Test Case
Validation and Verification Tool

STEP 5: Compare Test Case Results against Master
Validation and Verification Tool

STEP 5: Compare Test Case Results against Master
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

Simulator of Systems to validate that requirements meet the Design intention

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
Results

VALIDATION

REQUIREMENT VALIDATION
DEFECTS IDENTIFIED and RESOLVED

Function #1, 67
Function #2, 57
Function #3, 21
SYSTEM VERIFICATION
DEFECTS IDENTIFIED and RESOLVED

VERIFICATION
What’s Next?

Validation
Concept $\rightarrow$ MIL
Emphasis: Readability
Output: Requirements

Implementation
MIL $\rightarrow$ SIL
Emphasis: Efficiency, Compliance
Output: Software

Verification
SIL $\rightarrow$ HIL
Emphasis: Testability, Traceability
Output: Prototypes
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          MathWorks
Dr. Darrel Recker Ford Motor Company
Brad Hochrein    Ford Motor Company

Comments and Feedback can be directed to:
Nate Rolfes
nrolfes@ford.com