



FCA and CNH Industrial
Official Global Partners



MILANO 2015

Model based design at CNHi

The transition to full model based implementations

Carlo Cloet – Karel Viaene, CNHi Zedelgem

MathWorks Benelux user's conference

June 11th, 2015

Contains confidential proprietary and trade secrets information of CNH Industrial. Any use of this work without express written consent is strictly prohibited.



Key Takeaways

- CNHi is adopting an end-to-end model-based development strategy
- Model based design is a key enabler for developing innovative functionalities

CNH Industrial

Our Products



Trucks



Buses



Firefighting Equipment



Civil Protection and
Defence Vehicles



Skid Steer Loaders



Crawler Excavators



Engines
and Transmissions



Tractors



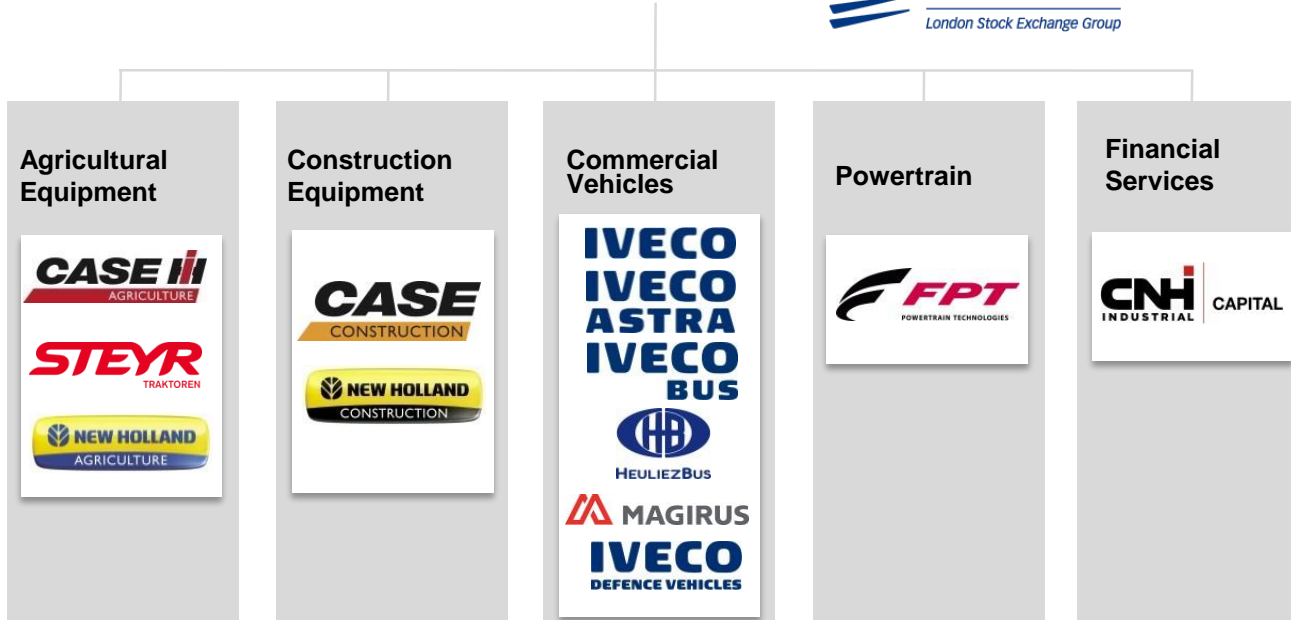
Combines

CNH Industrial

Company Structure



&



CNH Industrial

Key Figures (31 Dec 2014)

1

company

12

brands

49

R&D Centers

64

plants

190

national markets

69,207

employees

6,100

individuals dedicated to innovation

7,518

active patents owned

\$ 1,106

million invested in R&D

\$ 6

million invested in training

\$ 708

million net income

\$ 32,555

million revenues

\$ 8,857

million total available liquidity

Note: all figures provided herein are on a US GAAP \$ basis unless otherwise indicated

Model based design at CNHi (Zedelgem)

+10 years of model-based design experience



Display Logic



Forage harvester automated filling



Sugar Cane Harvester Drivetrain



Grape Harvester Drivetrain



Combine header height control
Hydro-pneumatic suspension
Feedrate control



Baler HIL Testing

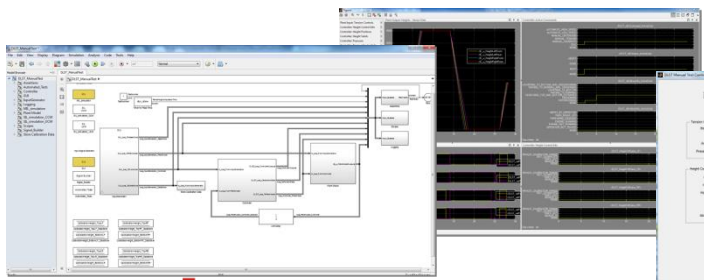


Baler application logic

Innovation Challenges and Achievements

Transitioning to a complete model-based implementation

- Objective: convert combine legacy C code to a full model-based implementation
 - Simplify transfer from innovation to product development
 - Avoid manual integration work
 - Consolidate legacy code base
 - Increase level of abstraction
 - Leverage MIL capability (test early, test often), rapid prototyping, etc



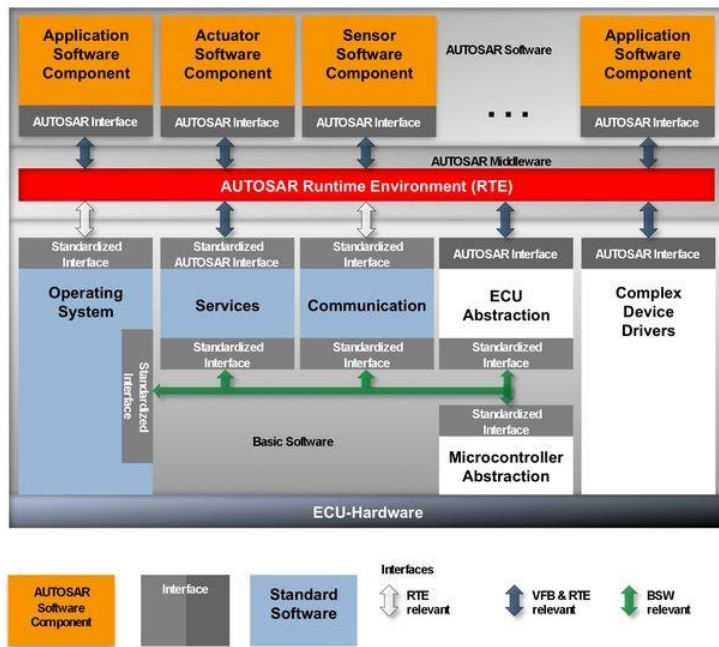
How did we get there and leverage MathWorks?

- 3 major challenges:
 - System integration
 - Team based development
 - Functional implementation and validation

How did we get there and leverage MathWorks?

Challenge 1: System Integration

- How to integrate 100's of models into a functioning vehicle?
- Key enablers: AUTOSAR architecture + Simulink built-in AUTOSAR support

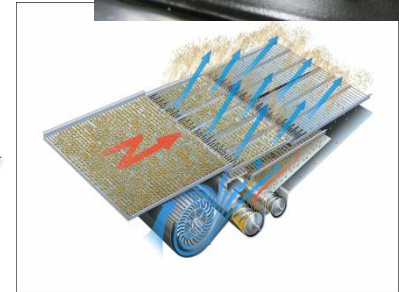
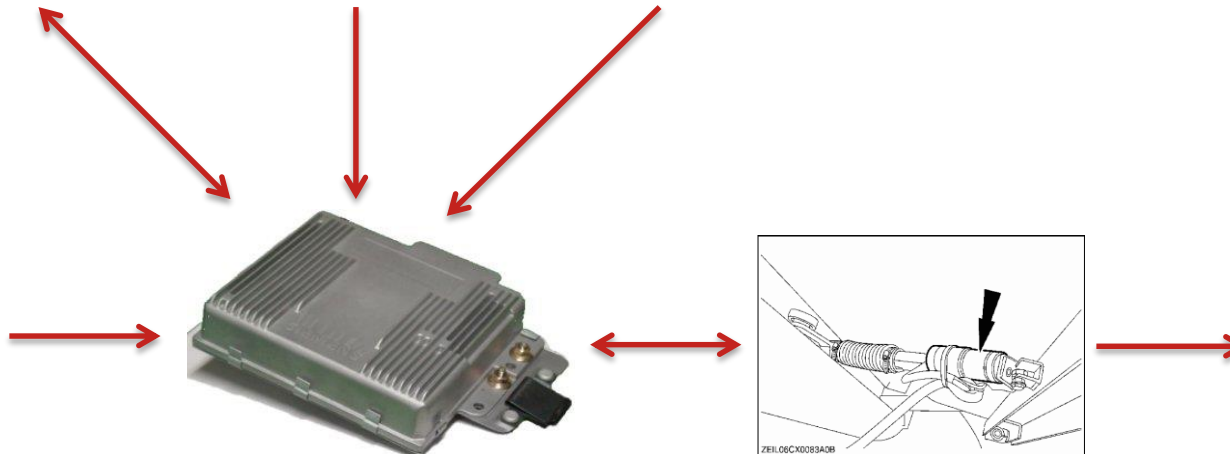
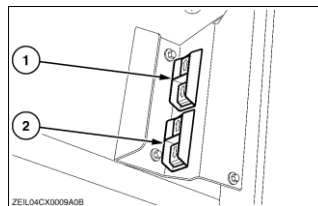
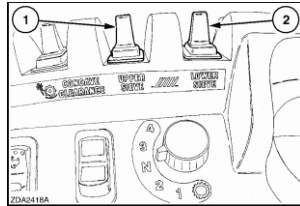
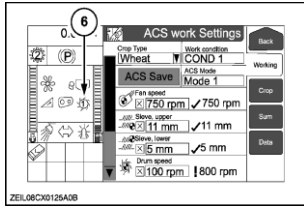


Source: <http://www.autosar.org/about/technical-overview/>

How did we get there and leverage MathWorks?

Challenge 1: System Integration

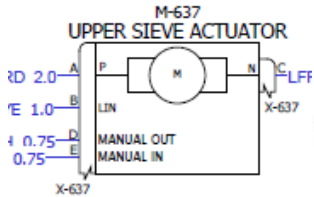
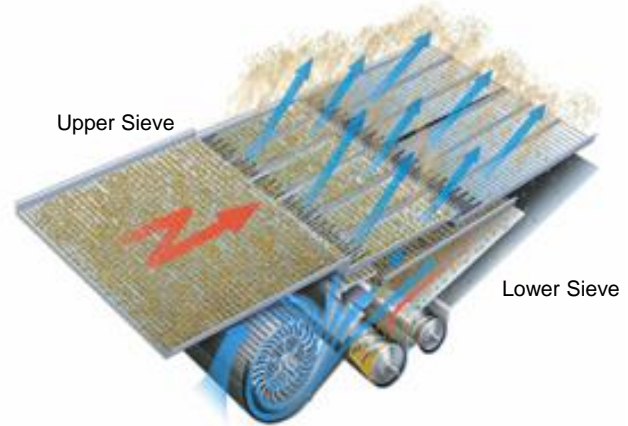
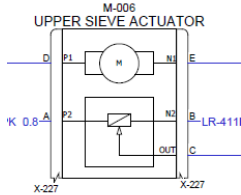
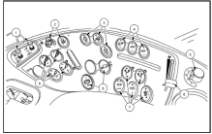
- Example subsystem: sieve control logic



How did we get there and leverage MathWorks?

Challenge 1: System Integration

- Additional integration challenges



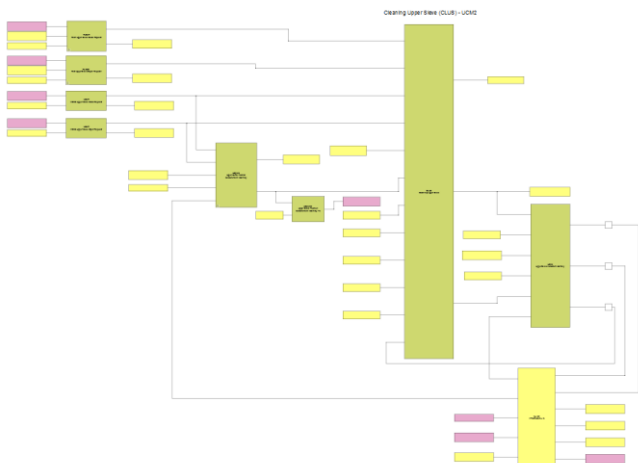
Same functionality,
different hardware

Same functionality,
multiple instances

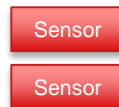
How did we get there and leverage MathWorks?

Challenge 1: System Integration

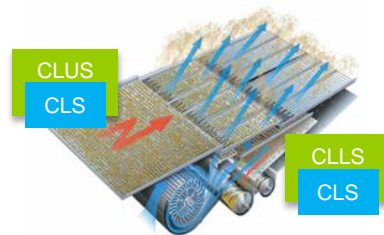
- Approach: component based architecture



Application SWCs
Sensor SWCs
Actuator SWCs



Application SWC re-use
by use of engineering unit interfaces



Multi-instance support
via Simulink library implementation



How did we get there and leverage MathWorks?

Challenge 2: Team based development

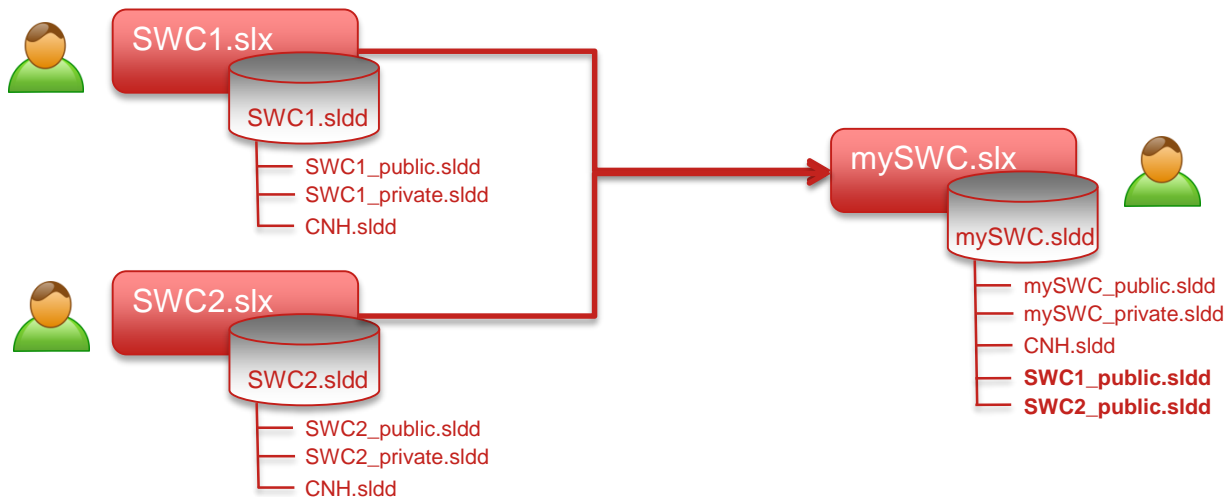
- How to enable concurrent model development by multiple people?
- Key enabler: Simulink data dictionary feature (introduced in R2014a)

The screenshot displays the Simulink Model Explorer interface. On the left, the Model Hierarchy shows the project structure with 'DLST' selected. The main pane shows the 'Dictionary Objects' for 'DLST', listing various objects such as 'DLST_busCalibReply_t', 'DLST_busSuspensionHeights_SuspendedTracks', and 'DLST_busTracksCalibValues_SuspendedTracks'. On the right, a 'Data Dictionary: DLST' panel provides 'Information for: DLST', including the file path, creation and modification dates, and a list of 'Referenced Dictionaries' like 'CNH (195)', 'DLST_public (14)', and 'INDIAG3_public (3)'.

How did we get there and leverage MathWorks?

Challenge 2: Team based development

- Using references to “public” data dictionaries to obtain input signal definitions
 - Public data dictionaries establish contracts among otherwise independent developers



How did we get there and leverage MathWorks?

Challenge 3: Functional implementation and validation

- Intellifill on Forage harvester
 - Forage harvester operation
 - Automated trailer filling
 - ✓ 3D camera
 - ✓ <https://www.youtube.com/watch?v=-zaQnygsMuQ>
 - Why automation
 - Challenges





How did we get there and leverage MathWorks?

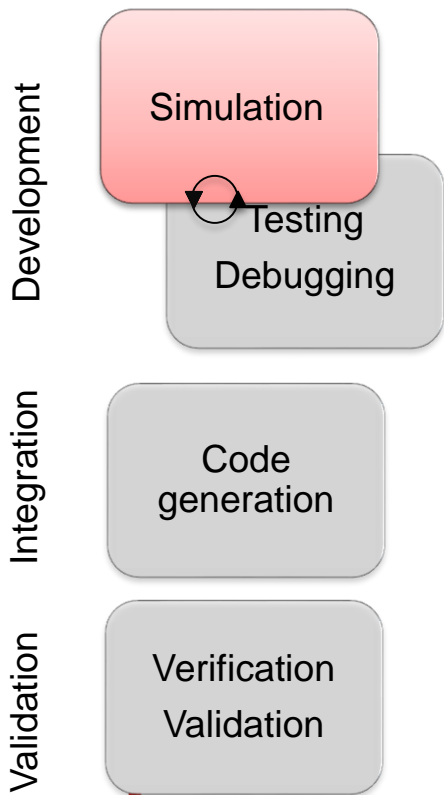
Challenge 3: Functional implementation and validation

- Intellifill on Forage harvester
 - Forage harvester operation
 - Automated trailer filling
 - ✓ 3D camera
 - ✓ <https://www.youtube.com/watch?v=-zaQnygsMuQ>
 - Why automation
 - Challenges

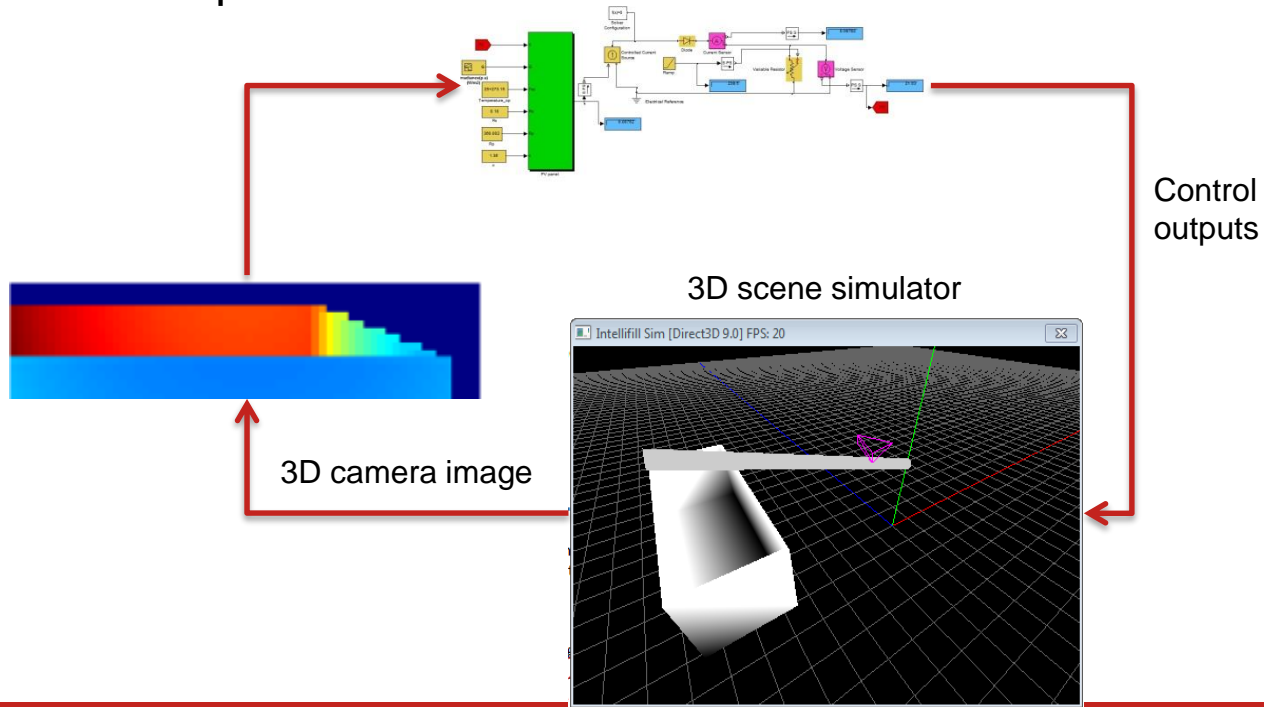


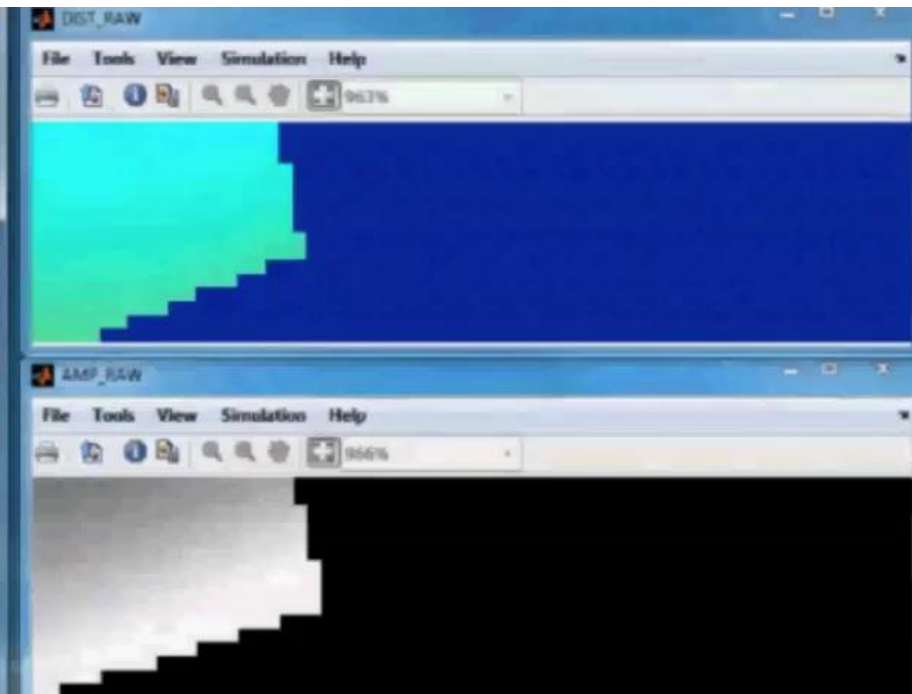
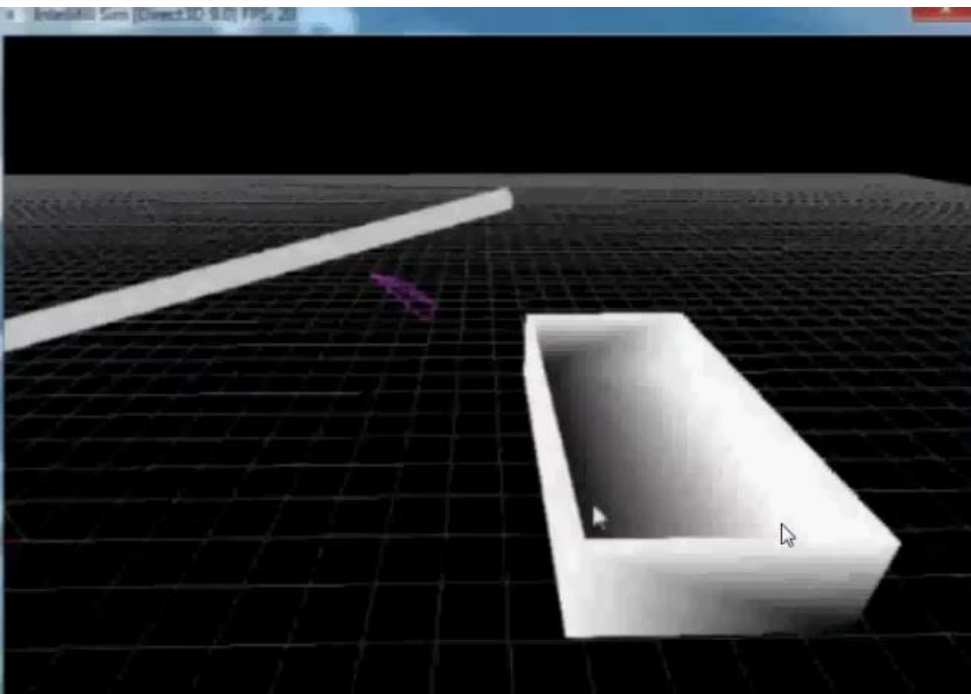
How did we get there and leverage MathWorks?

Challenge 3: Functional implementation and validation



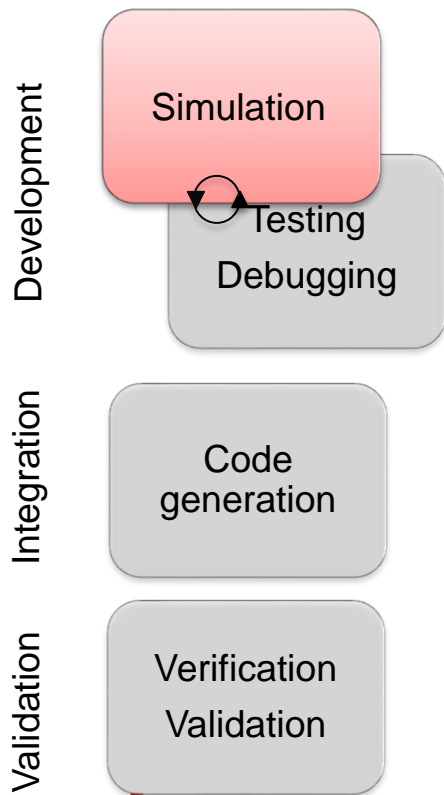
- Software architecture and algorithm development through closed loop MIL simulation



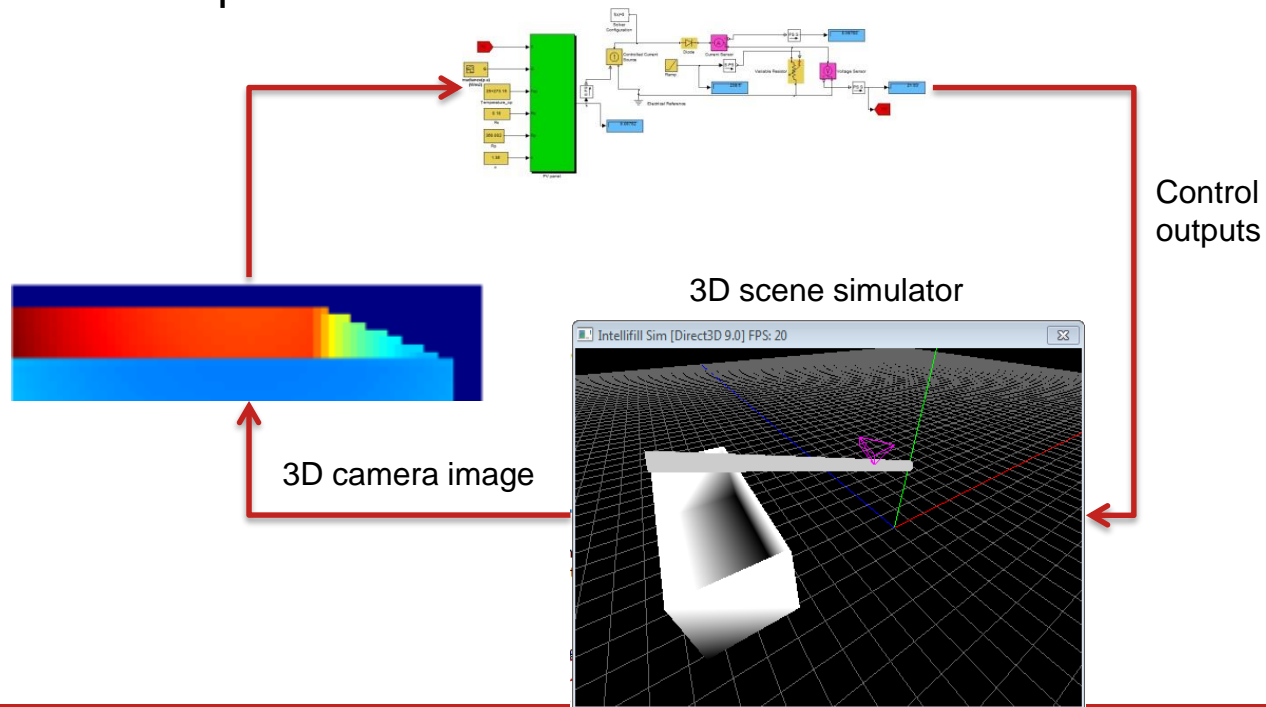


How did we get there and leverage MathWorks?

Challenge 3: Functional implementation and validation



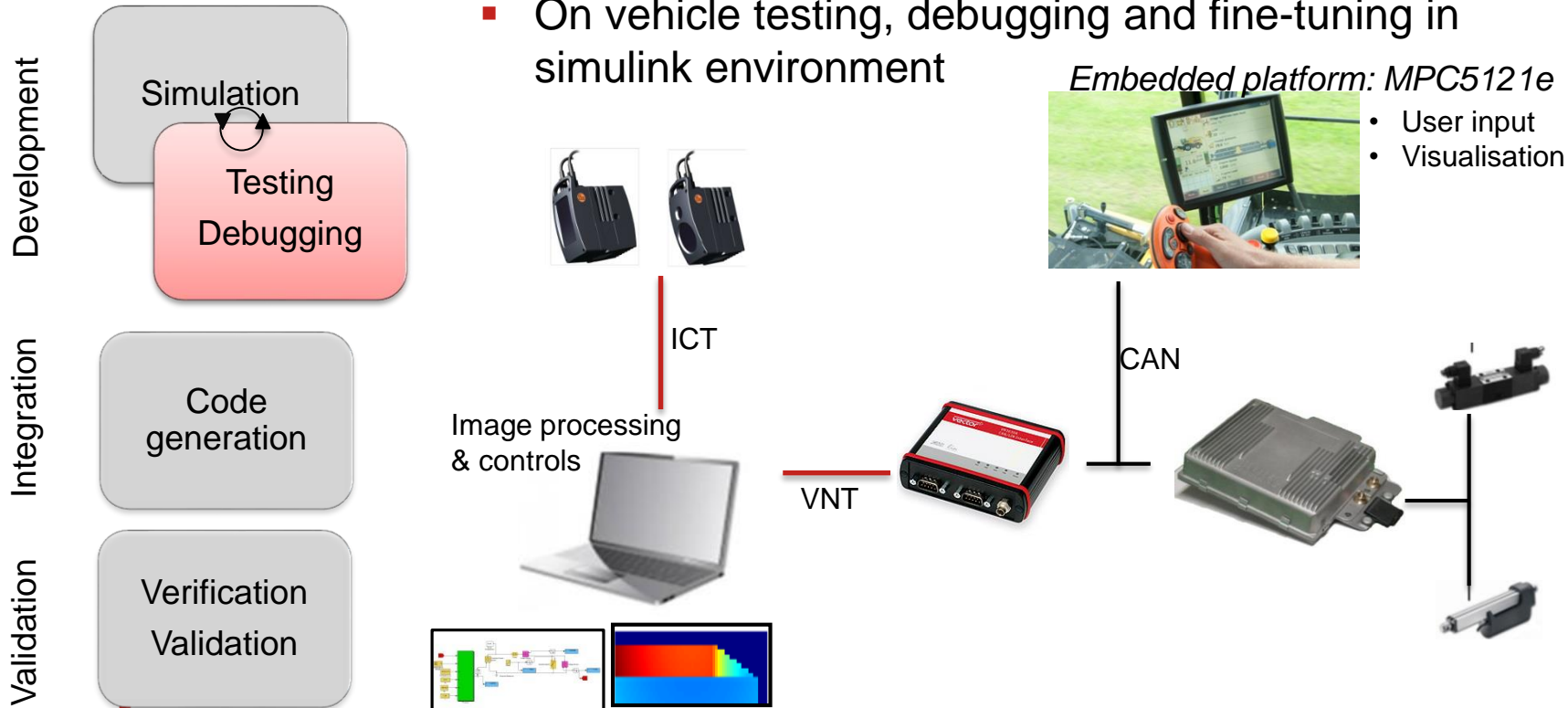
- Software architecture and algorithm development through closed loop MIL simulation



How did we get there and leverage MathWorks?

Challenge 3: Functional implementation and validation

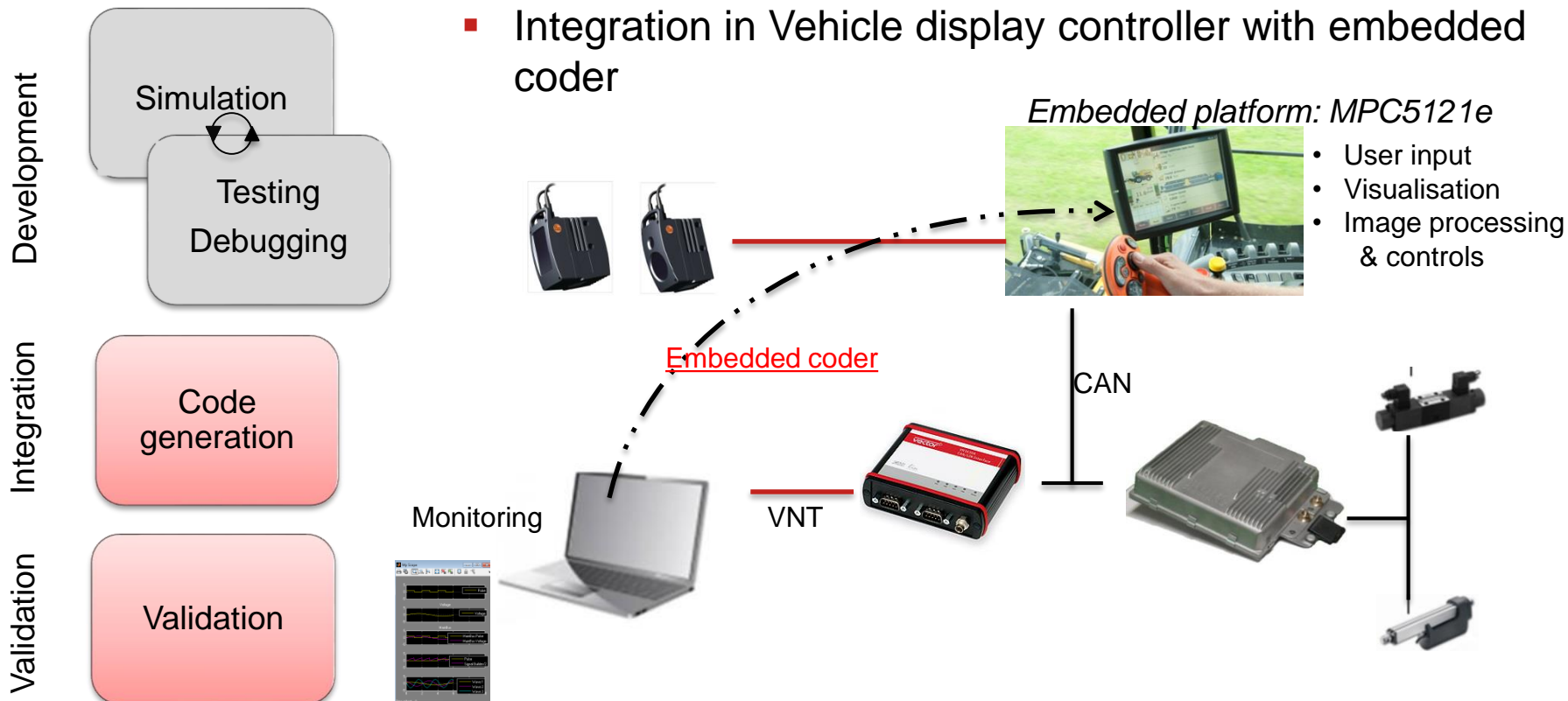
- On vehicle testing, debugging and fine-tuning in simulink environment



How did we get there and leverage MathWorks?

Challenge 3: Functional implementation and validation

- Integration in Vehicle display controller with embedded coder



Key Takeaways

- CNHi is adopting an end-to-end model-based development strategy
 - Enabler 1: Simulink built-in AUTOSAR support
 - Enabler 2: Simulink datadictionary feature
- Model based design is a key enabler for developing innovative functionalities
 - Enabler 1: MIL simulation: Closed loop simulation with 3D scene simulator
 - Enabler 2: Rapid Prototyping: Instrument control toolbox & Vehicle network toolbox
 - Enabler 3: Code generation: Embedded coder allows fast integration