The Role of Real-Time Workshop® Embedded Coder in Supporting Cummins Inc. Vision for Model Based Development

MathWorks Automotive Conference
June 19-20, 2007
Dearborn, Michigan
Agenda

- Introduce Cummins
- Analysis Led Design Strategic Initiative
- Vision For Embedded Controls Development
- Meeting the Need:
  - Model Based Development
  - Product Line Architectures
- Challenges
- Why Real-Time Workshop® Embedded Coder?
- Future of Real-Time Workshop Embedded Coder
- Conclusions
Cummins is a Worldwide Diesel Engine Leader

- Cummins at a glance (2006)
  - Engines & power generation since 1919
  - 33,500 employees
  - $11.4 billion in sales revenues
- Engines cover a wide range of applications
  - Engines to equip from Dodge RAM SUV to 200 tons mining truck
  - Heavy-duty trucks, buses, marine equipment, and power generation
- Cummins serves customers in over 160 countries
Why Analysis Led Design at Cummins?

- Cummins is a major producer of diesel engines in the automotive, industrial, power generation, & marine markets.

- Cummins is under constant pressure to deliver excellent fuel economy, emissions compliance, & durability at the lowest possible cost.

- EPA emissions mandates are driving increases in the complexity of controls’ algorithms, and hence, the embedded controller code.

- Analysis is key to getting the highest quality products into customer’s hands as rapidly as possible - “cut and try” is not efficient.

- Even after careful analysis, Cummins requires tools & processes to allow the more efficient development & testing of new designs.

- Anything Cummins can do to streamline & dovetail all phases of the V-cycle helps reduce cycle time and error.
Cummins’ Vision for Embedded Controls Development

Cummins generates models of the controller... to produce automatic C-code straight from the controller model...

& test these models in simulation... that is tested on actual hardware which is >95% bug free.
Meeting the Need

- Model Based Development (development side):  
  - Create system design environment  
  - Design, develop, & analyze a controller for managing a plant  
  - Generate C code automatically from Simulink® diagrams

- Product Line Architecture (software side):  
  - Realize similar core software throughout multiple product lines  
  - Define interfaces to allow addition/deletion of SW without compromising system integrity (plug-&-play)
Merging of Model Based Development & Product Line Architecture

MBD: Solution to Controller Complexity WITHIN a Product Line

PLA: Solution to Controller Complexity ACROSS Product Lines

Automatically Generate Embedded Code

Produce Embedded Controller

Without Sacrificing Simulation Capability
Challenges

● Product Lines are currently not pervasive throughout industry.
● Current 3rd party code generation tools are not based on this paradigm → but they are getting there!
  ● Biggest roadblock is less about code generation & more about simulation:
    ● e.g., function calls → how to realize in simulation?
    ● e.g., pointers → how to realize in simulation?
  ● Modifications are needed to allow current tools to meet requirements.
● Embedded controller design spans many different organizations within the company.
  ● Workflows are different
  ● End products are different
● Burden of proof rests with showing/proving benefits.
Why a Code Generator Now?

- Our workflows, existing tools & current processes foster the divergence of models → This divergence will only grow!

- Industry is adopting Model Based Development methods to design controllers.

- New code generation tools are becoming capable of meeting our architecture requirements.

- There is a synergy to do this now & it is a corporate technical productivity strategy.

2002 controller was complex. 2007 is more complex. 2010 is even more complex.
Embedded Controller Development
V-Cycle

**Goal:**
- Single Simulink model of controller for entire V-cycle

**Bridging the Gap to Reach Goal:**
- Synergistic embedded controls culture
- Fully integrated tool base

*Bridge-the-Gap*
Why Real-Time Workshop Embedded Coder?

- MATLAB® and Simulink® are already workhorses of Cummins controls development organization.

- The MathWorks has demonstrated a superior support environment.

- Real-Time Workshop Embedded Coder . . .
  
  - Links code generation directly to simulation
  
  - Provides greater flexibility in the design of controller models
  
  - The MathWorks pilot team provides support packages allowing the “bending” of the tool to meet our needs.
The support packages help to meet our basic architectural needs.

Key SW parameters (throughput, memory, code size) are being monitored & early results look promising.

Targeted component models are being converted & integrated into our total system simulation modeling framework.

Model Based Development using Real-Time Workshop Embedded Coder is beginning to be embraced by upper management.

Process & workflow changes are being pushed throughout company.

Other tools are being modified to integrate with Real-Time Workshop Embedded Coder.
Future of Real-Time Workshop Embedded Coder

- **Cummins Key Initiatives**
  - PLA is key to success of software
  - MBD is key to success of controller algorithm development

- **Real-Time Workshop Embedded Coder must**
  - embrace PLA
  - as well as MBD
  - continue to foster a collaborative relationship with Cummins

Product Line Architecture

Model Based Development
Conclusions

- The design of embedded controllers has become more challenging due to EPA rules & regulations.
- Cummins has been pursuing Model Based Development for last decade.
- Our major challenge is closing the gap between development & code generation.
- Cummins selected Real-Time Workshop Embedded Coder to close this gap.
- Preliminary analysis shows great promise both in realizing simulation and in code generation.
- The MathWorks has given us excellent support so far.
- Real-Time Workshop Embedded Coder must continue to evolve to meet demands of SW created by Product Line architected systems.