Model Based Design in a Seamless Embedded Software Process

Craig Stephens
Chief Engineer
Controls Research & Advanced Engineering
Ford Motor Company
Success in Implementation of MBD

• Industry is broadly successful in deploying MBD at a feature level or for Rapid Prototyping, HIL, etc.

• This success now drives a focus on the Enterprise view of MBD in the context of a broader Model Based Systems Engineering Process.
Current State and Future Opportunity

• Model Based Design is sufficiently widespread to consider it the norm.
• Results show tremendous opportunity for delivering complex advanced functions with higher quality and without compromising on expected automotive delivery schedules.
• Deployment of MBD in large organisations and distributed systems continues to be challenging.
• The bigger opportunity lies in the integration of MBD Controls and SW Design into a fully integrated CAE Process

MBD Tools and Process still need work but the value added upfront far outweighs the risks.
Application of the Capability and Maturity Model Integration to MBD

- **Initial**
  - Depends on the competence and heroics of the people.
  - Inability to repeat successes

- **Managed**
  - Standardised processes used even under times of stress.
  - Whole organisation capable of repeatably producing the same output on time.

- **Defined**
  - Process consistently applied across projects.
  - Detailed measures of the process and work products.

- **Quantitatively Managed**
  - Quality and process performance understood in statistical terms

- **Optimised**
  - All aspects of process measured and optimised with respect to organisational objectives.

Increase MBD Capability:
- Adaptation of SW Engineering Tools/Methods to fit MBD.
- Reduce variation in output.
- Add process/product measurement capability.

Ref: CMMI Institute: http://cmmiinstitute.com/
MBD to Model Based Systems Engineering

Control Design with Simulation.

Model Based Rapid Prototyping

Model Based Design with Autocode

Model Based Systems Engineering

Enterprise Systems Engineering

Optimised

Quantitatively Managed

Managed

Initial

Individual Success

Enterprise Success
Complex and Complicated!

**Engine ECU (Per Model-Year)**

- 35 APPLICATIONS
- 150 FEATURES
- 200 DEVELOPERS
- 10,000 SIGNALS
- 30,000 CAL PARAMS

… and with changing Architecture, Interfaces, New Functions, Variant Functions…..
MBD System “V” Process Map Overview

Phase I
- Requirements and DVP

Phase II
- Model Development
- Calibration
- Code Generation

Phase III
- Application Functional Testing
- Integrated Application Functional Testing

Phase IV
- Model Documentation

Phase V

Phase VI

Phase VII

Phase VIII

Application Functional Testing
Model Documentation
Integrated Application Functional Testing
Requirements and DVP
Model Development
Calibration
Code Generation

Inputs
- Parameter Mgmt
- Architecture
- Test Vectors
- SW Installation Guide
- MBD Checklist for Strategists
- Style Guide Compliance Report

Outputs
-Ctl Mdl Proc Doc
-Blocksets
-Lib Utils
-Style Guide
-Parameter Mgmt
-Rapid Prototype Module

Phase VII

Phase V

Phase VI

Phase III

Phase II

Phase I

Research & Advanced Engineering
MBD Workstreams

- Documentation
- Version Control
- Change Control
- Architecture
- Blocksets
- Library Utilities
- Style Guide
- Parameter Management
- Calibration
- Variant Management
- Control Model Development
- Code Generation
- DV Testing
- Application Build
- RAM/ROM/Chrono/Tasking
- User Environment
- Code to Model Conversion
- Plant Modeling
- Management & Training

MBD Core Team drives common processes and practices across Ford
User Environment
Ford Automotive Modeling Environment

- A Ford Custom Tool that sits on top of Matlab.
- Part of a “standard” MBD SW load.
- Controls the use of approved Library Blocks
- Maintains a consistency of models across the program or multiple programs
- Automate model compliancy checking
- Integration to configuration control system
- Set up of environment achieves generation of efficient C code
- Allows for integration of specialized API’s
  - Interfaces to other software
  - Plug-ins for auto-test programs

FAME provides a consistent environment for the whole team that can eliminate common mistakes and provide automation for tedious, common or difficult tasks
Parameter Management
UniPhi: Complexity Visualization

VISUALIZE WHAT I HAVE ANY WAY I NEED TO SEE IT

UniPhi Server

e.g. What are all dependencies for this signal?
UniPhi Summary

- Centralized data management
  - Version control
  - Recreate history of objects, releases
- Interoperability: Share data across tools
- Centralized system architecture design and management
- Visualization of complexity
- Rule checking
- Roles & Permissions
- Support for C-code as well as models
- PLM interface

UniPhi provides a consistent enterprise-wide collaborative development environment
MBD Challenges

- Integration with the supply base:
  - Interface management (AUTOSAR)
  - Requirements & validation cascade.
- Building and maintaining plant models.
- Scalability and integration of MBD tools in large extended teams.
- Total process support (documentation, parameter management..)
- Support for different business models across the across our enterprise (outsourced vs. insourced, fixed vs. floating point, etc.).
- Validating new versions of the MBD Tools and planning migration.
- Autocode improvements:
  - Quick response to issues with both Interim Containment Actions and Permanent Corrective Actions
MBD in Context
Beyond Controls

MODEL BASED SYSTEMS ENGINEERING

MECHATRONICS

- Control Systems
- Digital Control Systems
- Control Electronics
- Electronic Systems
- Mechanical CAD
- Electromechanics
- Mechanical Systems


Source: wiki.hsc.com, Google images

Research & Advanced Engineering
Model Based Systems Engineering

- Directly connects the development engineers to Customer Experience and Vehicle Attributes.
- Moves from a component Design and Release focus to a process around system dynamics.
- Enables the Controls Engineer to interact upfront with the hardware engineers.
- Facilitates data migration, integration and integrity.
- Provides *necessary* visibility and access to *relevant* data.
Tying it all Together

Hardware Design & Implementation
- Mechanical
- Thermal
- Manufacturing process
  CAD

Software/Controls Design & Implementation
- Control Algos
- SW & Calibration
- ECU & Multiplex
  Matlab
  PLM/PDM
  MBD
  Cal files

Requirements Development, Analysis, & Testing
- Regulatory, Attribute, Functional
- Balancing & Trade-offs
  PLM/PDM
  DVMs
  Design Rules

Functional Safety Analysis
- Safety Goals
- HARA
  STPA
  ISO 26262

Systems Model
- Requirements – EoO
- Use Cases
- Behaviors
  - Functions
  - Flows
  - Architecture (hw/sw)
    - Interfaces
    - Views
  SysML

CAE Modeling & Analysis
- Plant & Control Models
- Simulation/Co-Simulation
- Static & Dynamic Analysis
  AMESim
  Simulink
  Dymola

Failure Mode Avoidance
- P-Diagrams
- RRCLs
- FMEAs
  FMA Tools
  PLM/PDM

Requirements
- Requirements
- Functions
- Device Transmittals
- Behaviors
- Data Results
- Data Sources
- Lessons Learned
- Regs & Interfaces

Arch Options
- Constraints & Capabilities
- IC EF, PM
- Risk Analysis

Research & Advanced Engineering
Summary

• A state of efficient Model Based Design with Autocode is in sight.
• Entering an exciting phase that will enable future complex systems by:
  – Maturing the MBD Controls & SW Process.
  – Connecting Engineering information across Mechanical, Electrical, Control & Software.
  – Connecting engineers in distributed organisations.
  – Allowing all the stakeholders to contribute to successful system execution.
Thank You!