Pragmatic Strategies for Adopting Model-Based Design for Embedded Applications

Eric Dillaber– Principal Technical Consultant
The MathWorks, Inc.
Introduction

• **Large volume of MBD literature**
  • Companies share processes & benefits
  • Vendors provide seminars on new features
  • Best practices
  • No shortage of information

• **How to get started?**
  • Initial adoption is a significant challenge for most
  • What capabilities to initially target?
  • How to transform current process to achieve gains & minimize risk?

• **Goal:**
  • Tap collective experience
  • Look at what the most successful companies have done
  • Identify common themes and collect a set of proven strategies
Model-Based Design

- What’s MBD?
- Why do it?
  - Make Models?
    - Make products!
  - Eliminate HW prototypes?
    - Minimize HW prototypes!
  - Build it right the first time?
    - Build it wrong a thousand times!
- How to get started?
The Phased Approach to Adoption

- **Phase 1**: Proof of Concept
  - Initial Migration Plan

- **Phase 2**: Deploy Component
  - Deploy Full Application

- **Phase 3**: Initial MBD Process
  - Plan
  - Assess
  - Execute

- **Phase 4**: Deploy Enterprise-wide
  - Optimize & Improve
Components, Applications and the Evolution of an Embedded Application
Timing the Phased Approach

- **Plan**
  - Phase 1: 3-6 mo
  - Proof of Concept
  - Initial Migration Plan

- **Execute & Refine**
  - Phase 2: 5-9 mo
  - Deploy Component
  - Deploy Full Application

- **Optimize & Improve**
  - Phase 3: 1-3 yr
  - Initial MBD Process
  - Deploy Enterprise-wide

- **Phase 4**
  - Continuous
Phase 1: Proof of Concept

Theme: Discovery
- Discover features/capabilities* of the tools and how to address your needs with them

Activities:
- Identify Objectives/Capabilities
  - Capacity for Complexity, Productivity, Quality?
- Train
- Model
- Execute on Target

Key Output & Results:
- Demonstrate feasibility, build support
- Build experience and capability
- Initial Migration Plan

Considerations:
- Keep it simple
  - Algorithmic complexity
  - Existing function & build
- Industry stats:
  - 60% introduced in spec
  - 55% detected in testing
- Dedicated Cross-functional team

Common Pitfalls:
- Production deliverable
- Expecting ROI
- Aggressive timing without help
The Migration Plan

- Objectives
- Metrics
- Organization
- Training
- Process Changes
- Constraints
- Standards
- Automation

This plan will change – it is not static!
Phase 2: Deploy a Component

**Theme:** Production Component
- Apply/refine new capabilities in production

**New Activities:**
- Executable Spec
- Model-Based V&V
  - Simulation-Based Testing
- Integration w/Production Build
- Integration w/Config Management

**Considerations:**
- Often most challenging reqmts
- MBD for component only
- New skill sets

**Common Pitfalls:**
- Complexity, Size, and Representation
- Outsourcing migration*
- Graphical coding*

**Key Output & Results:**
- Production component delivered
- V1.0 MBD process definition
Phase 3: Deploy an Application

Theme: **Scale Up**
- Develop the full application using MBD

**New Activities:**
- Requirements linking
- Model-Based V&V
  - Coverage, SIL, PIL, etc
- Automation
  - Standards checking, integration/build, testing

**Considerations:**
- Multiple organizations affected
- Choosing what to model
- Standardized MBD environment

**Key Output & Results:**
- Production application delivered
- Significant return on investment
- V2.0 MBD process: Base set of capabilities

**Common Pitfalls:**
- Architecture scalability*
- Touching the code*
- Lack of modeling standards
Phase 4: Optimize & Improve

Theme: **Continuous Improvement**
- Fully leverage success: Rollout to other programs, dept, divisions
- Optimize: Improve upon

New Activities:
- Assess other site requirements
- Refine objectives & capabilities

Key Output & Results:
- Replicated success at multiple sites
- Dramatic productivity improvement
- Increased capacity for complexity

Considerations:
- Continuous Improvement: Tools & products are constantly evolving

Common Pitfalls:
- No dedicated central tool group*
- Rollout w/o adequate representation
Pragmatic Strategies for Adopting Model-Based Design (SAE Paper 2010-01-0935, Dillaber, Kendrick, Jin, Reddy)

Strategies to consider in planning your phased approach:

- Assess organizational challenges and impact
- Plan for change
  1. Identify the problem you are trying to solve
  2. Choose a project with proper complexity and technology
  3. Mitigate risk with a phased approach
  4. Choose the appropriate legacy components for migration

Strategies to help you choose what areas and capabilities to target first
(Selected key items)

1. Use executable spec development as an opportunity to solidify requirements
2. Make the model a source for documentation
3. Choose architecture and component technology early
4. Establish and enforce design standards
5. Develop a plant model with “trend-correct” behavior
6. Verify what you need, not what you want
7. Migrate key supporting processes such as CM
GM Standardizes on Model-Based Design for Hybrid Powertrain Development

**Challenge**
Develop new hybrid powertrain technology for GM vehicles

**Solution**
Standardize on MathWorks tools and Model-Based Design for control systems design and production code generation

**Results**
- Aggressive delivery date met
- Worldwide collaboration and communication enabled
- Designs reused across product lines

---

“The Two-Mode Hybrid powertrain took Model-Based Design to a new level within GM. This project provided the confidence and experience we needed to apply MathWorks tools for Model-Based Design on other large-scale global engineering programs.”

Kent Helfrich
General Motors

Link to user story
Example 1
General Motors Powertrain

2003 Project Results

- Some modifications to the GMPT Algorithm Modeling Standards were required to support auto-code
- Some modifications to Software Process / Coding Standards were required to support auto-code
- Efficiency of the automatic code generation tool should not factor in the code
- Auto-code was successful for the ECU controller

2005 Automatic Code Generation Objectives

- Re-evaluate hand code process and auto-code process to reduce overhead
- Address automatic code generation for multiple large functions in a single model file
- Develop a seamless process for testing the model and the generated code
- Develop on-target rapid prototyping process using Mathworks Embedded Coder
- Improve usability of the automatic code generation process
- Deploy first automatic production code build in actual production powertrain vehicles in Q1 2005
- Rollout production code generation to many additional users and production programs

Pilot Result - Metrics

<table>
<thead>
<tr>
<th></th>
<th>Hand Code</th>
<th>Auto-code</th>
<th>Percent Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration ROM</td>
<td>9464</td>
<td>9464</td>
<td>0%</td>
</tr>
<tr>
<td>900</td>
<td>900</td>
<td>350</td>
<td>-1.76%</td>
</tr>
<tr>
<td>938</td>
<td>38</td>
<td>938</td>
<td>-0.82%</td>
</tr>
</tbody>
</table>
Lear Delivers Quality Body Control Electronics Faster Using Model-Based Design

Challenge
Design, verify, and implement high-quality automotive body control electronics

Solution
Use Model-Based Design to enable early and continuous verification via simulation, SIL, and HIL testing

Results
- Requirements validated early. Over 95% of issues fixed before implementation, versus 30% previously
- Development time cut by 40%. 700,000 lines of code generated and test cases reused throughout the development cycle
- Zero warranty issues reported, 12 months post-production

"We adopted Model-Based Design not only to deliver better-quality systems faster, but because we believe it is a smart choice. Recently we won a project that several of our competitors declined to bid on because of its tight time constraints. Using Model-Based Design, we met the original delivery date with no problem."

Jason Bauman
Lear Corporation

Link to user story
Example 2
Lear Corporation

- Identified the problems to solve & tracked w/metrics
- Phased in capabilities focusing on ROI first
- Integrated with existing development process
- Automated and optimized after process integration
Next Steps?

- Read the paper
- Review the MBD literature
- Connect with others who have gone through the process before
Thank You!