2009-01-1145

Adoption of Modeling Standards as a Part of Enterprise-Wide Deployment

Mirko Conrad
Michael Burke
The MathWorks

Copyright © 2009 SAE International

ABSTRACT

The successful deployment of modeling packages across large organizations depends on the adoption of a uniform set of modeling standards within the organization. Further, the automation of modeling standards enforcement greatly facilitates their adoption. This paper examines several case studies of large-scale deployment of Model-Based Design and the areas where standards enforcement automation was employed. The paper further examines several areas where automation could be employed to improve the deployment process.

INTRODUCTION

The modern software development process is an iterative design task involving multiple individuals and groups. In some cases, vendors outside the company will work on the software component. Handoff of the software component between participants in the design process is facilitated by following an accepted style guideline. (Note: This paper uses the terms *style guideline* and *modeling standards* used interchangeably. Further, the term *modeling standards checking* refers to the process of verifying a model's correctness with respect to a modeling standard.) This paper examines the successful practices for facilitating the adoption of style guidelines across an organization.

BENEFITS DERIVED FROM USING ENTERPRISE-WIDE STYLE GUIDELINES - The basic roles of modeling guidelines and the benefits derived from their adoption are as follows:

- Consistent visual presentation
  - Facilitates understanding the model
  - Ensures readability of the model
- Uniform interfaces
  - Reduces integration problems
  - Facilitates exchange of models
- Validation of (critical) configuration settings
  - Ensures expected code generation
  - Enforces expected model/block behavior
  - Facilitates traceability

For these reasons, many formal processes, such as IEC 61508-3 and MISRA-C AC, highly recommend the use of modeling standards.

METHODOLOGY - This paper is based on interviews and conversations with personnel in automotive companies who have based their modeling standards on The MathWorks Automotive Advisory Board (MAAB) Style Guidelines. The paper draws on the observation of processes in three automotive customers, with secondary observation from four additional automotive/commercial vehicle equipment companies.

The information from these customers has been synthesized to extract the common best practices. Illustrative examples have been pulled from each of the companies as appropriate.

OBJECTIVE - This paper presents the process steps required for successful adoption of modeling standards in an enterprise-wide deployment. Case examples of the benefits derived from following the process steps are provided; further, when possible, the problems that led to the adoption of the process step are discussed.

REQUIREMENTS FOR SUCCESSFUL ADOPTION OF MODELING GUIDELINES

This section of the paper is divided into six sections, each covering one aspect of the guideline adoption process. Although they are presented as discrete tasks, there is overlap between the tasks. The rollout process is, by its nature, a highly iterative task in which the adopting company adapts guidelines and methods in response to its user community.

GUIDELINE MANAGEMENT - Successful deployment of a set of guidelines requires a group dedicated to the management, education, and automation of the modeling standards process. Their tasks include:

- Selecting guidelines

- Developing the guideline enforcement process and creating a process document
- Creating automation routines associated with guideline enforcement
- Educating end users
- Maintaining guidelines and automation routines

The guideline management group should either include members from or receive input from all major groups who will be using the guidelines. In several of the companies interviewed, the initial set of guidelines were selected without end user input. The resulting requirements were considered burdensome. Additionally, some of the checks were considered inappropriate to the task or stage in development by the end user.

SELECTION OF GUIDELINES - For four of the six interviewed organizations, the selected guidelines were derived from an existing set of published modeling guidelines; the remaining two companies initially used in-house developed rules but are currently moving toward the adoption of a published rule set. In all cases, the companies did not use the full set of rules from the published source; instead, they sub-selected the rules based on several factors, such as:

- Development stage (e.g., initial, HIL, RPC, production code generation)
- Project/target (e.g., safety critical, fixed point, legacy modules)
- The end user's role (e.g., researcher, calibrator, software engineer)

The companies interviewed are using or adopting the use of the MAAB Style Guidelines as their base set of rules. The MAAB Style Guidelines comprise more than 80 rules covering many aspects of the model's design and function. The typical company selected between 30 and 40 of the MAAB rules to enforce with an additional 10 to 15 company-specific rules, resulting in a total set of between 40 and 50 rules.

To prevent the rule set from becoming a burden, companies adopted several methodologies. The methodologies directly relate back to the guideline selection factors listed at the start of this section.

<u>Development Stage and Guideline Compliance</u> - Part of the challenge of selecting the guidelines was determining at what stage in the development which guidelines would be enforced. A common theme, a near universal solution, was to have loose compliance in the initial stage with full compliance required at the time of the final handoff of the model.

In general, the early stages of development emphasized the interface and visual aspects of the style guidelines. In the later stages of development, the block and model configurations, especially those related to code generation, were added into the required guideline compliance.

Another way of looking at the stages is that in the early guideline, enforcement was targeted toward facilitating integration and readability of the model. The later stages target the functional and code generation aspects of the model.

The primary benefit of staged enforcement is to reduce the burden of model correction on any one engineer. The initial design engineers fix the interface/visual prior to handing off the model to the release engineers, who then just need to be concerned with aspects of the model related to code generation and correctness.

<u>Project/Target and Guideline Compliance</u> - Most companies used a project- or target-based criterion for modifying the base set of guidelines. Projects and targets are just one way of thinking about the required modifications to the rule set; examples include:

- Fixed-point processor (target)
- Nonvolatile (NV) memory requirement (target)
- Safety critical (project)
- Transmission group versus engine group (project)

Examples of the type of rules that are modified or added include enforcement of:

- Fixed-point checks (fixed point)
- Data storage class requirements (NV memory)
- Redundancy of signal processing (safety critical)
- Different naming convention for data (transmission versus engine)

(The naming convention example is noted as an example where having input from the end users results in a smoother processes. Because the rule group did not recognize the two *different* sets of rules used by the transmission and engine groups, the initial rule set did not include a naming convention. Based on feedback from the transmission and engine groups, each was assigned a unique set of naming convention rules.)

<u>End User's Role and Guideline Compliance</u> - The end users' role directly affects the level of compliance to which they will adhere and the type of corrective operations they are required to perform. The key factor is that users are only supposed to fix errors in their area of expertise. Examples of role-appropriate corrections include:

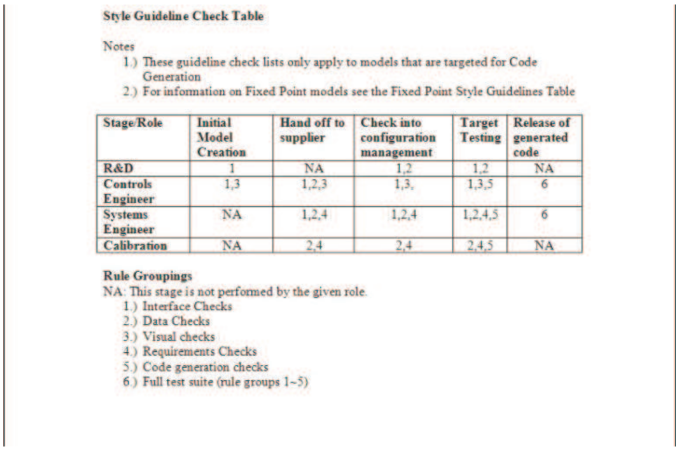
- Calibrators: Data-related errors
- Software engineers: Interface and scheduling
- Researchers: Functional correctness

PROCESS AND PROCESS ENFORCMENT - The process document specifies when guideline verification must be preformed (See Figure 1). The common points for testing for all companies were:

- At check-in to configuration management system
- Prior to production code generation

Additionally, some companies run:

- Prior to running test suites
- Prior to HIL/RPC code generation
- Prior to design reviews
- At initial model load



Style Guideline Check Table

Notes
1.) These guideline check lists only apply to models that are targeted for Code Generation
2.) For information on Fixed Point models see the Fixed Point Style Guidelines Table

Stage/Role	Initial Model Creation	Hand off to supplier	Check into configuration management	Target Testing	Release of generated code
R&D	1	NA	1,2	1,2	NA
Controls Engineer	1,3	1,2,3	1,3,	1,3,5	6
Systems Engineer	NA	1,2,4	1,2,4	1,2,4,5	6
Calibration	NA	2,4	2,4	2,4,5	NA

Rule Groupings
NA: This stage is not performed by the given role.
1.) Interface Checks
2.) Data Checks
3.) Visual checks
4.) Requirements Checks
5.) Code generation checks
6.) Full test suite (rule groups 1~5)

Figure 1. Process compliance template.

EDUCATION OF END USERS - Every organization stressed end-user education. Three primary areas were recognized as being significant to successful adoption by end users:

- Knowing how to follow the process. End users need to know when and how to start the guideline checking process.
- Understanding the significance of the selected checks. Since in some cases the guidelines impose a burden on the end user, an understanding of why the checks were selected is important.
- Knowing how to correct errors detected by the process.

The average instructional period was a one hour class supplemented with a user's document and / or online help. The quality of the error messages (i.e. the information returned to the user in the case of a failure) had a direct impact on the need for training.

USE OF AUTOMATION IN THE PROCESS - Two aspects of automation are relevant to the guideline deployment process. The first is guideline compliance verification, or checking. Check automation verifies that the model complies with the conditions set down in the guideline documentation. Given that models frequently contain thousands of blocks, manual verification of compliance would be laborious.

The interviewed companies all use some form of check automation, either in-house or using a third-party tool. Most companies expressed interest in using commercial off-the-shelf tools such as the Model Advisor in Simulink from The MathWorks.

The second key automation point is at the process gateways. Having a mechanism that automatically launches the model compliance verification at the process gateways ensures that the end user completes these steps.

In one case where the company used a manual launch of the checks prior to model check-in, the estimated compliance rate was only 35%. After an automated launch of the checker on model check-in was implemented, compliance levels reached an estimated 90% (the end user could bypass the launch of the checker).

REPORTING METHODS - There was general agreement among companies about the need for a compliance report that:

- Has a human- and machine-readable format
- Is in a format that can be added to configuration management
- Has a summary section
  - Overall pass/fail information
  - Version information
  - Date/author information
- Has a detailed section for guidelines that were not in compliance

The reports (see Figure 2) were used both by the automation process (in the machine-readable format) and by the end user (for design reviews and daily work).

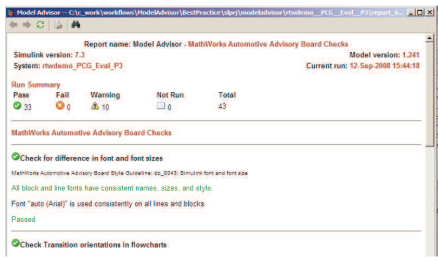


Figure 2. Summary report.

PDF and HTML reports were generally considered acceptable formats for the human-readable reports. The machine-readable report was generally based on a MATLAB data format since the automation tools were based in that environment. Machine-readable formats include CSV, M-file, and Excel.

The machine-readable format was tied back into the automation processes. It was used to ensure that gateway steps were completed before end users continued with their process. Failure to use a machine-readable format and relying on user reporting resulted in a percentage of the end users not complying with the guidelines. (For the two companies that originally did not use machine-readable reports, an estimated 10% and 15% of the detected failures were not reported.)

DEPLOYMENT

At all of the companies interviewed, the deployment process was iterative. An initial set of guidelines and automation processes was deployed to a test group.

At all of the companies, the test group was either an “advanced development” group or a “production” group. These types of groups were selected because:

- Their models had the most rigorous compliance requirements.
- They represented the majority of the end users.
- They had the greatest need for automation of requirements (because of tight time budgets, they required full automation).

The test phase lasted three to five months with three to four iterations in both the automation and rule set. During this time period, bugs in the automation process and missing or suboptimal guidelines were addressed.

The two most successful methods for expanding deployment were:

- Add-on training: New groups adopted use after completing training on the tool.
- Incremental compliance requirement: The tool was rolled out to the full company; however, end users were given a six-month time period with increasing compliance requirements.

Company-wide deployment without training and at full compliance resulted in the guideline support group being overwhelmed by help requests and subsequent dissatisfaction in the end-user community.

CONCLUSION

There was widespread agreement among the companies involved in the study about the importance of guideline adoption. Company-wide adoption of modeling standards was highly dependent on the acceptance of the process by the end users. The acceptance, in turn, was dependent on the early involvement in the definition of the compliance and automation processes.

Interviews with multiple companies uncovered six key process areas that were crucial to the successful adoption of standards. The overall deployment strategies highlighted the need for process validation and user education for successful adoption of the style guidelines.

REFERENCES

1. Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow – Version 2.0. www.mathworks.com/industries/auto/maab.html, The MathWorks Automotive Advisory Board, Style Guidelines Working Group, July 27, 2007.
2. Simulink Verification and Validation Product Description. www.mathworks.com/products/simverification/description6.html, The MathWorks, 2008.
3. H. Röbig, A. Leicher, T. Klein, T. Farkas, M. Born, J. Zander-Nowicka, Werkzeugübergreifende Konsistenzsicherung von Artefakten bei der Entwicklung softwarebasierter Systeme im Automobil. Workshop Automotive Software Engineering (ASE’06), Dresden, Germany, 2006.
4. Begic, G., Checking Modeling Standards Implementation. *The MathWorks News & Notes,* June 2007.
5. Mirko Conrad, Ines Fey, Hartmut Pohlheim, eGuidelines – A Tool for Managing Modeling Guidelines. International Automotive Conference, Detroit (US), 21–22. Juni 2005, www.mathworks.com/industries/auto/iac/presentations/conrad_paper.pdf (June 2005).
6. T. Erkkinen, Model Style Guidelines for Production Code Generation. SAE World Congress 2005, Detroit, MI (US), March 2005 (SAE Techn. Paper Series #2005-01-1280).

DEFINITIONS, ACRONYMS, ABBREVIATIONS

MAAB: MathWorks Automotive Advisory Board

Check: When the validation of a style guideline has been implemented in an automated format

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.