



How Small Engineering Teams Adopt Model-Based Design

The benefits of adopting Model-Based Design are well documented for large organizations. An interesting fact from MathWorks experience is that Model-Based Design always starts with a small engineering team, regardless of the size of the company. And, like the large organizations, the initial small team used Model-Based Design to reduce development time and costs.

This paper outlines the successful adoption practices and considerations that small teams used.

What Is Model-Based Design?

Model-Based Design provides a mathematical and visual approach to develop complex control and signal processing systems. It centers on the use of system models throughout the development process for design, analysis, simulation, automatic code generation and verification.

Engineers create a model to specify the behavior of an embedded system. The model, which consists of block diagrams, textual programs, and other graphical elements, is an executable specification that lets engineers run simulations to test ideas and verify designs throughout the development process.

Teams can benefit from having a system model serve as the specification:

- The design can be tested, refined, and retested throughout the development process.
- It's easy to try new ideas because there's no need to build prototypes.
- Testing and validation are done continuously rather than at the end of the process so that many errors are found and corrected before hardware testing.
- Embedded code can be generated automatically from the system model, which reduces effort and eliminates hand-coding errors. This code can be used for testing real-time simulations.
- Models can be adapted and re-used on subsequent projects.

Getting Started with Model-Based Design

Small teams cannot afford to be slowed down by changes in the way they develop embedded systems. In practical terms, this means that any new tools and process changes need to be introduced incrementally, essentially on the fly. Trying new approaches and design tools carries an element of risk. Successful teams have mitigated this risk by introducing Model-Based Design gradually. They usually start with a single project, identifying early wins that can be achieved using Model-Based Design versus using the current practice.

Successful introduction of Model-Based Design involves taking incremental steps that can help a project along without slowing it down:

1. Experiment with a small piece of the project. A recommended approach is to take a new area of the embedded system, build a model of the software behavior, and generate code from it. This is a small change and one that a team member can make with a minimal investment in learning.

This will demonstrate that:

- High quality code is created without hand coding.
 - The code matches the behavior of the model.
 - A model can be simulated to work out the bugs in the algorithms much more simply and with greater insights than dynamically testing C code on the desktop.
2. Build on initial modeling success. The next valuable activity to adopt is system-level simulation. Typically, once a segment of code has been shown to work well, the developer will incorporate it within the rest of the application and discover hardware-software integration issues. (These are the same issues one would find if all the code is hand built.)

By contrast, system-level simulation allows the engineer to perform system integration virtually. To do this, an engineer needs to create an adequate model of the hardware in the system and the basics of the environment around the system. Model fidelity should be high enough to ensure interfacing signals have the right units and are connected to the right channels, and the right level of dynamic behavior is captured to test the algorithm.

The results offer an early view of how the hardware and the embedded software will behave. This is especially valuable if the hardware is in development and does not yet exist or is expensive to prototype.

3. Use models to solve specific design problems. Engineers can gain targeted benefits even without developing full scale models of the plant, environment, and algorithm. For example, suppose a team needs to decide on the required parameters for a solenoid used for actuation. They can develop a simple model that draws a conceptual “control volume” around the solenoid, including what’s driving it and what it’s acting on. The team can test various extreme operating conditions, and derive the basic parameters without having to derive the equations. Through simulation, engineers develop the forces needed and then feed them back to the hardware design activity. This model can then be stored for later use to solve a different design problem or for use in the next product development project.
4. Stick with the basics. For small teams, the immediate benefits of Model-Based Design include the ability to create component and system models, use simulations to test and validate designs, and generate C code automatically for prototyping and testing. Later, they can consider advanced tools and practices include modeling guidelines, automated compliance checking, requirements traceability, and software build process automation. However, for small teams, these can be addressed at a later
5. Leverage the experience of MathWorks. MathWorks engineers have worked closely with large and small work groups globally. They have years of experience helping small teams apply Model-Based Design to reach an end goal. A good first step for a small team interested in adopting Model-Based Design would be to request a free guided evaluation to help the team come up to speed quickly.

Success with Model-Based Design

By adopting Model-Based Design incrementally, small engineering teams have consistently achieved immediate and tangible results:

- Faster time to first demonstration
- Faster time to market with a high quality product
- Expanded capacity for developing complex systems

Small teams depend on the efforts of engineers who put in long hours, show high dedication, and move quickly. With an incremental approach, these teams can smoothly adopt Model-Based Design and perform at even higher levels of speed, competence, and design quality.