Massachusetts
Institute of
Technology

Model-Based Systems Engineering: Documentation and Analysis

Week 4: Managing the Model

Project

Name

Tomas Mawyin

Instructions

Before you begin, you should save your project on your local drive. We recommend the following format:

Lastname_Firstname_Course3_Week4

Please note: You will not be able to re-download your file after submission; therefore, please keep this file in a central location for future reference.

The work in the project deliverable is **individual**.

After you submit your project, you will self-assess your work as well as the work of three peers. If you have any questions, feel free to contact a TA in the Discussion Forum.

Although work is strictly individual, sharing ideas and concepts with other students is encouraged.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location. Be sure to submit your assignment at least one hour before the deadline to provide time for troubleshooting.

Once the deadline passes, you will not be able to upload the document and therefore will not be able to submit and complete the assignment.

Week 4 Project

Overview

Your assignment for this project is to build a model management plan for an MBSE system. You are encouraged to use the same system that you selected in Weeks 1 and 2.

As you learned in Week 4, there are many important factors to consider when designing a model management plan.

Develop your model management plan and touch on eight key components.

REQUIRED STEPS

Model introduction

Step 1: Initial verification and validation of the model

Step 2: Governing the inputs to the model

Step 3: Communicating model results

Step 4: Model configuration management

Step 5: Defining the model owner

Step 6: Funding the model

Step 7: Process for model changes over the lifecycle

Step 8: Model end of life and renewal planning

Submit your project and review others' submissions

Model Introduction

Briefly describe your MBSE approach for which you are creating a model management plan.

I am modeling the Electronic Power Assist Steering (EPAS) system. I would be using a ground-up approach for this project. This is because currently there are no available models for the EPAS system.

Currently, the existing approach is to separate the steering group into mechanical, electrical, and software teams. Each team work on creating requirements documents that will address the main design. For example, the mechanical team will create requirements to meet loads, durability, and geometry of the system. On the other hand, the software team creates requirements to meet the controllability of the vehicle at different speed. This process happens with some interaction between teams but it is very difficult to understand where common faults might happen or how requirements should be cascaded among teams. The MBSE approach aims to address the issue of change propagation and help in the creation of requirements because there would be a common repository of information.

Using MBSE can also avoid having to create several documents when describing different aspects of the system. In the existing approach, each team develops their own documentation and there is always issues with version control. MBSE would reduce the amount of work teams needs to do when developing these documents and by extracting the information from the source, there won’t be an issue with version control.

Finally, the MBSE approach can provide flexibility during integration with other features. Many software features are already modeled and with a modeling approach it would be easy to drop/remove features for upcoming projects.

Step 1: Initial Model Verification and Validation

How do you validate the model in the first place? Under what bounds does the model need to be valid? Who will validate the initial model?

Model validation can be done by using existing data and running already established test cases for the steering system under various conditions. By doing this, the subject matter experts can have a better picture of how the system behaves during the integration of the mechanical, electrical and software subsystems.

The initial model should be validated by the model curator which has a high-level expertise in all aspects of the steering system. The bounds of the model are giving by the requirement at higher loads, meaning that the system should be able to steer the vehicle under the maximum load conditions.

Step 2: Governing the Inputs to the Model

Will restrictions be applied to each of these inputs? How will bugs and flaws in the model be reported and addressed?

Model inputs should be controlled by the model owners. Each model owner should restrict the inputs to the model to a given range based on the parameters established in the model scope. For example, smaller vehicles require less load ranges that larger vehicles. Inputs to the model, should then be restricted to other uses in the form of a GUI, where users can only select inputs within the specified range.

For updates or bugs, the users will have a management system (typically Github issues or Jira) where the model owners and the model curator can discuss and solve the issues. The model owner will have the authority to assign issues to developers on their teams but any consolidation to the model should be approved by the model curator.

Step 3: Communicating Model Results

Will the assumptions and caveats be communicated along with the results? If so, how? Will the supporting test cases be communicated along with the results? If so, how?

Model results should be available to all team members (including management). It is expected that the model curator or model owners would develop the required documentation from the model to record results at each vehicle milestone. All results should be complimented with the proper evidence – meaning that all results for the different test cases should be included.

Finally, it would be required that the results be presented to a panel of technical experts (preferably those experts not involved in the project) to avoid biases and catch possible error states.

Step 4: Model Configuration Management

How can multiple projects use the same model? Will data restrictions be imposed for different project teams?

All core models will be kept in a global repository. It is expected that developers would instantiate each model and adjust it depending on their project. Each project would have a common workspace where all the models would be connected.

Data should only be assigned to a particular project. This is particularly important because each project represents a particular vehicle variation and there should be no two identical variations in the organization.

Step 5: Defining the Model Owner

What responsibilities does the model owner have (technical, financial, etc.)? What decision rights do they have?

Model owners are in charge of the following tasks:

- Secure financials required to develop their model in their area of expertise
- Ensure all the technical expertise and manpower is provided by the organization
- Ensure all the development team has the proper rights to access the model
- Ensure that the core model is up to date
- Discuss with other model owners in the project about linkages and interfaces between models
- Work with the model curator to fix any issues on the model

Step 6: Funding the Model

How will the model be funded?

Model funding would come from the product development branch of the organization. They are in charge of providing the required funding for each particular vehicle variation and therefore, any project that uses MBSE.

Note that funding conversations should include model owners as they understand what overhead is required for the development of the model

Step 7: Process for Model Changes Over the Lifecycle

How will changes to the model be re-validated? Who will re-validate them? Is there a need to show traceability to past results? How will they be handled?

Model changes would be re-validated by the development team. The model owner in charge must assign a developer to re-run the testing plan including all test cases relevant to the proposed change.
The developer should then communicate the results by providing traceability to pass results. This will ensure that any changes do not greatly affect the system behavior or design in a negative way.

In case of an invalid or negative change, the subject matter experts must provide a contingency plan on how to improve the design. Once the plan is approved, the model owner and developers must implement the changes and run the re-validation process once again.

Step 8: Model End of Life and Renewal Planning

What is the life cycle duration or time horizon for the model? Who is scanning outside to see what new modeling capabilities may be available?

All models will be kept after the project has finished for at least 10 years. This ensures that any recall or warranty issue can be traced back to the model and the design. The model instances and workspaces are kept in the vehicle project repository and must not be reused.
If a similar vehicle is to be produced by the organization, the model can serve as a reference only. For new projects, a new model should be instantiated from the core models repository and configured for the new vehicle project.

Submit Your Project and Review Others’ Submissions

Submit your completed Week 4 Project file

- Note: The maximum file size that can be submitted is 10MB.

Assess the work of your peers

- A scoring rubric can be downloaded from the Week 4 Project Instructions page.