

Model-Based Systems Engineering: Documentation and Analysis

Week 1: What Is MBSE?

Project

Name

Jane Doe

Instructions

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

Lastname_Firstname_Course3_Week1

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 PowerPoint, you will self-assess your work. 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 1 Project

Overview

Choose a system on which you will propose an MBSE approach. This system can be anything you want, although it will likely be more useful for you to choose something from your work. Please ensure that the system has at least a medium or high level of complexity -- such as a car, satellite, enterprise server, or open-source software.

If you are working in a team, one person in the group needs to be a domain expert for the model that the group selects.

After you have chosen a system, answer the questions in the following slides.

REQUIRED STEPS

Step 1: Choose system and define scope

Step 2: Define MBSE approach

Step 3: Define MBSE purpose

Step 4: List major tenets of MBSE

Step 5: Identify the most important qualities of great models

Step 6: Identify systems engineering tasks

Step 7: Submit and self-assess your project

Step 1: Choose System and Define Scope

What is your system? What is in scope for the MBSE effort for your chosen system?

For this project I am going to try to apply the MBSE to build a software system for an automated teller machine (ATM). An ATM is a complex system that has a lot of different components. It has a card reader then reads the credit/debit cards of the user. It has a UI that interacts with the user and allows him/her to check account details and withdraw cash. It also communicates with the bank to get the user's information and to process his financial transactions. Apart from this there are many other sub components to ensure security, transaction compliance etc.

The purpose of this MBSE model is to help manage the development of this complex system. In particular, we want to manage the requirements of all the different subsystems, identify any conflicts, and manage the code development. Furthermore, the model should ensure that all operating conditions and boundary conditions are met by the software, including the financial regulations and compliance requirements. The MBSE strategy will maintain a single source of information, incorporate the basic validations, and facilitate information flow across different tools used by teams.

Step 2: Define MBSE Approach

How would the approach to this project change with MBSE?

One of the biggest challenges for the project is to allow information flow between different teams such as software development, hardware/firmware teams, UI designer, testing and product management. This will be facilitated by the MBSE approach. The MBSE model will connect with software tools used by different teams and facilitate information flow. Therefore, if the product teams make a change to the requirements in the product management tool, then those changes will be communicated instantly to the development team and the testing team by updating the software that those teams use for this project. This will ensure that everyone has the same, up to date information at any given time. If there are any conflicts, then teams can mutually decide and roll back the changes if required. To facilitate this flow our MBSE approach needs to tackle three key challenges:

1. How information will be communicated from one team to another. As mentioned earlier, teams use different tools. The MBSE model needs to integrate with these tools and update them whenever there is a change. There are two parts to enable this. The first part is to monitor these tools to check if users are making any updates. If they have made updates and those updates satisfy the basic sanity check mentioned below, then this information needs to propagate to all the tools in the format that the different tools understand.
2. Once the MBSE model connects to the various software tools involved we also need to define rules that govern how changes to one software affect changes to others. Different tools don't speak the same language so there needs to be a translator that can relay the information. This job will be done by the MBSE model, and is the biggest part of the MBSE approach.
3. MBSE models need to have some basic sanity checks that would be required for any change so that it does not create downstream issues. An example validation could be to check if all the components of the system are performing at least one task after the change is made. Another validation could be to check if the change is contradicting other requirements or features.

Step 3: Define MBSE Purpose

What are the purposes of your MBSE effort? Describe the financial and non-financial benefits you expect.

An ATM deals with financial transactions and therefore the tolerance for errors is pretty low. Organizations face two very big challenges.

1. They cannot be very flexible about updating requirements later in the development cycle since it might have adverse consequences. Therefore they have to be very rigid, and sometimes they miss out on releasing a lot of cool features that they could offer to customers.
2. They have to spend a lot of money testing the software because the tolerance for error in financial transactions is very low.

Both these challenges translate to lower profits, either in the form of increased costs or reduced revenue. The MBSE approach tries to address both challenges. It creates better visibility through proper information flow. Therefore, organizations can make better tradeoff decisions when dealing with change requests or requirement updates. In addition, that reduces testing costs significantly because MBSE performs validations and enables automation.

Step 4: List Major Tenets of MBSE

Describe how will you model the system. Briefly describe your approach to each of the major tenets of MBSE.

Central Model or Federation of Models:

We are trying to group the outputs of various software used by the different teams because each software performs a series of complex functions that could be treated as a model in its own right. All these models connect with the MBSE model and people keep using the software tools that they are using today. Therefore, our strategy is a federation of models.

Model Views

The MBSE approach will accumulate the information and notify each stakeholder whenever there is a change made by someone else in the system. These could be in the form of alerts generated using email and in-software notifications. In addition, the MBSE strategy will set software triggers or hooks into the various software tools to update them whenever it finds stale information in any of them. This way users will have the most up to date information.

Model repository or library

As mentioned before, our MBSE approach will work as a translator for various software tools. This translation function could be built out in a modular way, so that we can add and remove software programs per will. The translation rules for each software program will be different and can be stored in a library or repository.

Standards and Patterns

The MBSE strategy will rely on web services and restful API standards to communicate with various software components. It would utilize web hooks to listen to any changes made by team members. It will also use triggers to update the information in the relevant software. These are example software standards used for communications.

Model Checking: Logically Verifiable Rules or Tests

As mentioned earlier our MBSE approach will have certain sanity checks so that people don't make changes willy-nilly.

Ontology

For our software development system there are certain definitions and terminologies that are used throughout the product development organization such as features, bugs, user stories, technical specification, and code bases. These entities have defined relationships. The MBSE strategy will use these existing ontologies in its approach.

MBSE Methodology

As defined above, the MBSE approach will have four parts: a listener that will find when an update is made to a software component using webhooks, a translator that translates the input from one software to a format understandable by others, a trigger to update the other software components with the changes, and a sanity check on the updates to make sure they are consistent with the overall system.

Step 5: Identify the Most Important Qualities of Great Models

Reflect on the qualities of great models – what are the top three you are concerned about for your MBSE approach?

Availability of Interfaces: One of the key success factors for the MBSE approach is its availability of interfaces. Since we need to talk to other software tools, each potentially having different interfaces, our MBSE strategy should support most software standards for communication, if not all that are used currently.

Internally Consistent: One of the biggest goals of our MBSE strategy is to facilitate information flow and to maintain the most up to date and current information across the project. To achieve this the model needs to be internally consistent. If multiple software components are making changes at the same time, it should identify what changes are being made by which software and make updates accordingly. No one should be able to update stale information in the model accidentally.

Verification & Validation With Models: Our MBSE strategy is heavily reliant on our model to perform sanity tests to ensure that we always have a complete and consistent system. This quality is another important enabler for other qualities like model fidelity and credibility.

Step 6: Identify Systems Engineering Tasks

Systems engineering has a variety of different tasks depending upon its role in the organization such as interface management, change management, and to facilitate information transfer. Which of these or other tasks in your view are applicable to your chosen MBSE strategy?

In software development we don't really use the term systems engineering explicitly, however the tasks mentioned above are generally performed by software engineers/developers. In bigger projects where multiple development teams are involved, some of the developers have to work on interface management to enable interaction between two disjoint software components. Another role that sometimes developers need to deal with is change management, since they are the ones building the system, they often find issues and glitches in initial specifications during development and have to take the lead communicating these issues and getting them fixed from the right stakeholders. Both these tasks would be better supported by our MBSE approach since it allows for better information flow in the system.

Step 7: Submit and Self-Assess Your Project

- Submit your completed Week 1 Project file
 - Note: The maximum file size that can be submitted is 10MB.
- Assess your completed Week 1 Project
 - A scoring rubric can be downloaded from the Project Instructions page.