Paper ID: 35453
student

SPACE SYSTEMS SYMPOSIUM (D1)
Hosted Payloads - Concepts, Techniques and Challenges, Missions and Applications (7)

Author: Mr. Chrishma Singh-Derewa
National Aeronautics and Space Administration (NASA), Jet Propulsion Laboratory, United States

Ms. Kae Sawada
Caltech/JPL, United States
Mr. David Blamire
United States

A ROBOTIC TESTING FRAMEWORK FOR THE MODEL BASED ENGINEERING ENVIRONMENT

Abstract

Model Based Systems Engineering (MBSE) is an engineering methodology used for representing complex systems (such as launch vehicles) through visual modeling principles. MBSE captures functional behavior of asystem byusing models and diagrammatic descriptions to exchange data, allowing engineers to focus on the technicalities of their project rather than the document structures. The systems engineers at Jet Propulsion Laboratory have adopted this methodology due to its increase in efficiency, improved communication, and reduced risks and errors. Ensuring the proper function of the tools that enable this work is a high priority.

MBSE is implemented at JPL with the Engineering Modeling System (EMS), a toolset that provides the complete infrastructure needed to achieve the MBSE goals. The EMS contains a plugin that allow SysML data from a standard modeling tool like MagicDraw, to propagate to a Model Management System (MMS). The View Editor, auser friendly web portal, makes the data available from this server to users directly.A version of these components is available to the public through the Open Model Based Engineering Environment (Open-MBEE) project.

JPL has developed an end-to-end automated testing framework that ensures the integrity of the MBSE system beginning in the conceptual design phases, throughout the development, and in the later phases of mission integration. This testing infrastructure consists of several components. Java code automates actions in MagicDraw, including the synchronization of data with the MMS. Protractor automates actions that are performed on the View Editor.These components are combined into test scripts to ensure that the backend is communicating with either or both MagicDraw and the View Editor as expected. These scripts are then launched by the Robot Framework, which organizes and standardizes the test results into clear reports. Jenkins, a scheduling agent, automates the periodic running of these tests against development and release versions of the EMS.

Utilizing this testing framework,tests can be run regularly to ensure the stability and acceptability of the EMS implementation, and minimize the impact of code changes that introduce unexpected behaviors. Proper application of these tests allows developers to make changes and enhancements to the toolset without jeopardizing project or data integrity. This paper will detail the testing framework and its benefits by examining current and planned deployments, their associated results, and the impact on toolset development and deployment.