Applying a Model-Based Approach to Support Requirements Analysis on the Thirty-Meter Telescope

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

This article describes how a model-based systems engineering (MBSE) approach was used to support requirements analysis, system design, and early verification for critical subsystems of the Thirty Meter Telescope (TMT) [8]. The MBSE approach applied the Executable Systems Engineering Method (ESEM) [4] [5] and the Open-source Engineering Environment (OpenMBEE) [7] to specify, analyze, and verify requirements of TMT's Alignment and Phasing System (APS) and the Narrow Field Infrared Adaptive Optics System (NFIRAOS). The value proposition for applying this MBSE approach was to establish precise requirements and fine-grained traceability to system designs, and to verify key requirements using executable SysML [6] models beginning early in development.

□

Contents

Background
Purpose
MBSE Challenges
MBSE Approach
Methodology
Tooling
Analysis
Observed Benefits

Acknowledgements
References
Works Cited
Primary References
Additional References

Background

The TMT (Figure 1) is a next-generation ground-based extremely large telescope designed to answer key science questions regarding the nature and composition of the Universe. At its core is a wide-field, altitude-azimuth Ritchey-Chrétien design with a 492 segment, 30-meter diameter primary mirror (M1), a fully active secondary mirror, and an articulated tertiary mirror. Each segment's optical performance is sensitive to three rigid body degrees of freedom: piston, tip, and tilt. To obtain optimal image quality, the segmented M1 must perform like a single monolithic mirror, achieved through a multitude of controls working to co-align, co-focus, and co-phase its segments. The APS (Figure 2) is responsible for the overall pre-adaptive optics wavefront quality, using starlight to measure wavefront errors and align the TMT optics. Adaptive optics systems like the NFIRAOS (Figure 2) are designed to sense real-time atmospheric turbulence and correct the telescope's optical beam to remove its effect, enabling diffraction-limited imaging on the ground. In LGS MCAO and NGSAO modes, this is achieved through the use of wavefront sensors to detect laser and natural guide stars and deformable mirrors to direct the corrected wavefront to science instruments. These opto-mechanical designs and complex controls are constrained by several requirements that must be satisfied.



Figure 1. Thirty Meter Telescope. (Used with permission. Permission granted by Jamie Nakawatase.)

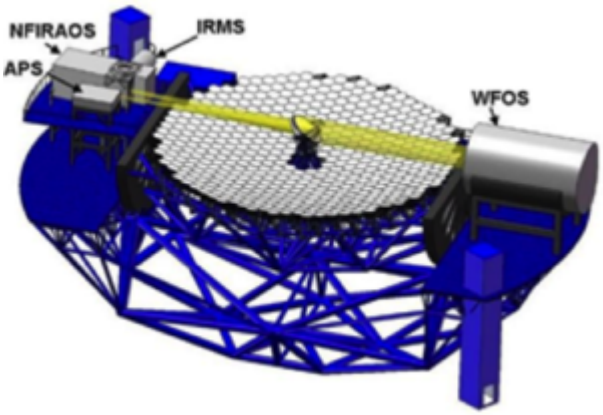


Figure 2. APS and NFIRAOS early light locations. (Used with permission. Permission granted by Jamie Nakawatase.)

TMT International Observatory (TIO), LLC is a non-profit organization of international members, responsible for managing the design, development, and operations of the TMT. The Jet Propulsion Laboratory (JPL) [9] participates in the design and development of several TMT subsystems and delivers an operational APS, where TIO is responsible for providing requirements to JPL. The APS team applies an MBSE approach to analyze requirements, derive an architecture design, and implement a system. TIO also works with JPL to analyze the operational behavior of NFIRAOS through modeling system-level operational scenarios (such as slew, acquisition, and dithering) with Monte-Carlo simulations. Modeling patterns are used to capture functional and physical system characteristics, behavior, requirements, parametric relationships, and use case scenarios. MBSE applications are motivated by optimization to better understand TMT's complex system behaviors.

Purpose

This article describes how MBSE is applied to the development of critical subsystems of a complex interdisciplinary system and the benefits of this approach. While document-based artifacts are necessary throughout the development lifecycle, complex systems engineering relies significantly on the use of models to address concerns from various domains (e.g. mechanics, optics, controls) (Figure 3). MBSE helps to manage implicit dependencies on information contained in these cross-domain documents, understand change impacts, analyze designs, and communicate evolving technical baselines. Models act as the single source of authority for systems engineering information that enables optimization through consistent and automated data exchange, enhanced analyses, and consolidating

subsystem design information into separate artifacts needed by various stakeholders.

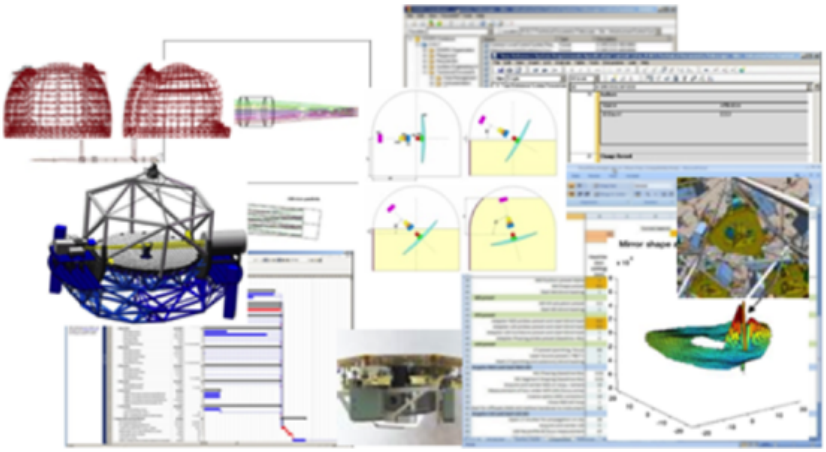


Figure 3. Landscape of Engineering Models. (Used with permission. Permission granted by Jamie Nakawatase.)

MBSE Challenges

Systems engineering (SE) is inherently challenging with concepts spanning several engineering disciplines. Expressing these concepts in a model demands the use of flexible methodology, language, and tools. The modeler must understand how to leverage this flexibility to rigorously specify systems with a broad range of complex design considerations. This poses the initial MBSE challenge—**the learning curve**. However, this challenge is native to any new undertaking and is overcome by hands-on experience, training, and ever-growing resources.

Another challenge is determining how to **apply** MBSE to meet the needs of a project. MBSE is not separate from SE—it is an approach to achieve the fundamental goals of SE more efficiently. MBSE should not be used to duplicate work, but instead replace efforts that can be better accomplished formally through modeling.

Engineers live in a landscape of variability among tools and information models in different domains (e.g. ALM, PLM, CAD), which are often implicitly connected [4]. To benefit from a model-based paradigm, the models require explicit connections. A key challenge is how to leverage data and associated models to enable **cross-domain integration**.

Finally, **standardization** is a challenge for MBSE. A model is only effective if stakeholders can understand it. SysML is an evolving modeling language that is becoming the dominant standard to communicate system architectures, independent of the modeling tools that

use it. However, SysML is only one of many standards that must be applied to ensure the models can be consistently interpreted by tools and users.

MBSE Approach

The MBSE approach follows the conventional SE V-process enriched by a model-based paradigm (Figure 4). The scope of this MBSE application addresses models at L2 and L3, and requirements flow-down to L4, although the approach can be applied to support specification and architecture design at other levels as well. Associated modeling artifacts are created early and maintained throughout the development lifecycle.

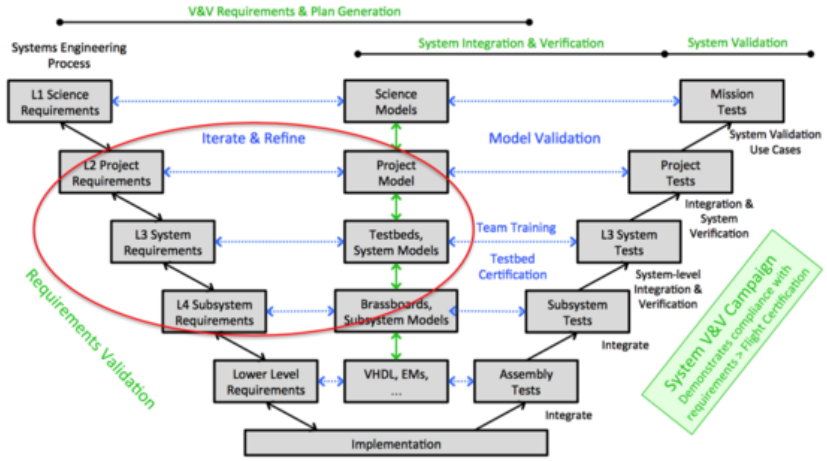


Figure 4. JPL Model-based V Process. (Used with permission. Permission granted by Jamie Nakawatase.)

MBSE articulates the system architecture in a formal, executable model that captures structure, behavior, requirements, and parametric relationships of system elements. Operational scenarios are defined in the model and analyzed through system-level simulation to verify requirements and validate overall system design over an expected range of conditions and parameters. The system model is also the authoritative source for several engineering documents. The MBSE approach is subsequently described by its methodology, tooling infrastructure, and analysis processes.

Methodology

The Executable Systems Engineering Method (ESEM) [1] is used to formalize requirements, specify system designs, characterize components, and specify/run analyses. ESEM augments the Object Oriented Systems Engineering Method (OOSEM) [2] by enabling executable models that enhance understanding,

precision, and verification of requirements through applying analysis patterns specified with various SysML diagrams. ESEM also enables integration of supplier/customer models.

Figure 5 shows the major activities common to SE processes using OOSEM. The red circles indicate where ESEM injects formal modeling methods.

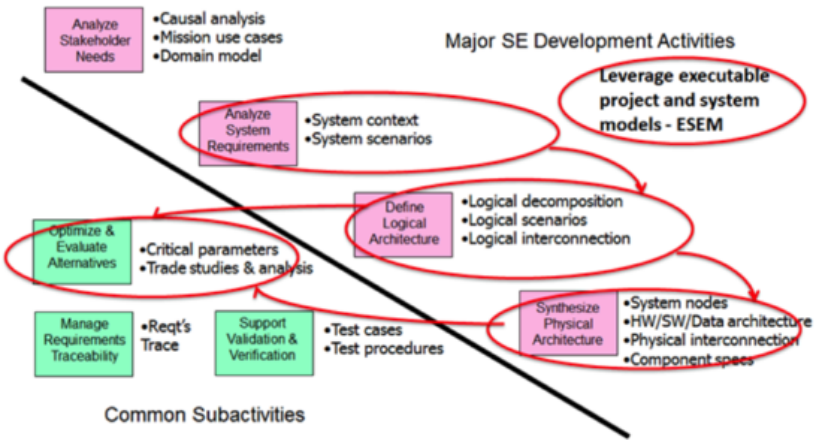


Figure 5. OOSEM activities. (Used with permission. Permission granted by Jamie Nakawatase.) [8]

ESEM is utilized to model different levels of abstraction that are analyzed using several modeling patterns as detailed in [4]. The system-of-interest is modeled as a black box that interacts with external subsystems, such as controls. Interactions are modeled using ports to identify operations and flows at the system-of-interest interface.

The conceptual model specifies technology-independent system components and captures their behavior. This part of the model is used to analyze characteristics such as duration of operational scenarios. Component behavior is captured using state machines and activity diagrams, and constraint parameters are captured in a table. Communication across internal and external system components is accomplished through the sending and receiving of signals through ports. This model supports production of interface control documents by querying information sent from one component to another over ports.

The conceptual model serves as a basis to specify the realization model of the physical components. The realization model imposes technology-dependent constraints on the design solutions. Both the conceptual (i.e. logical) and realization (i.e. physical) models represent the “as-specified” system.

Tooling

The MBSE approach uses standard SysML language and modeling tools to minimize custom software. The OpenMBEE community promotes an open tooling environment that provides a platform for modeling. It utilizes the Model Management System (MMS) that can be accessed from rich SysML desktop clients like MagicDraw, lightweight web-based clients like View Editor, computational programs like Mathematica, and other tools that utilize RESTful web services. The MMS also provides the basic infrastructure for search, relation management, versioning, workflow, access control, content flexibility, web applications support, web-based API access, and multi-tool/repository integration across engineering and management disciplines.

Figure 6 shows the integration of model artifacts produced by MMS, View Editor, and MagicDraw. System models are constructed, queried, and rendered following the view and viewpoint paradigm [3] from MMS.

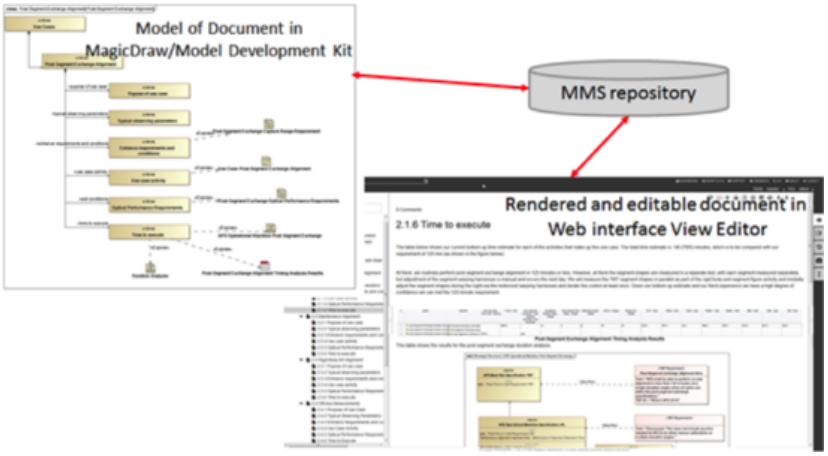


Figure 6. OpenMBEE interactions. (Used with permission. Permission granted by Jamie Nakawatase.)

Analysis

One key SE process is to analyze the impact of changing requirements on the system design. Figure 7 illustrates how the MBSE approach is used to support requirements impact analysis through the following steps:

- Step 1: A changed requirement triggers impact analysis.
- Step 2: MMS integrates DOORS (which manages text requirements) and the SysML model, enabling a DOORS requirement change to propagate to the SysML model.

- Step 3: A property-based requirement is formalized in SysML, enabling requirements specification that can be evaluated by engineering analysis.
- Steps 4-6: The conceptual and/or realization design is automatically verified against the changed requirement, resulting in pass or fail.

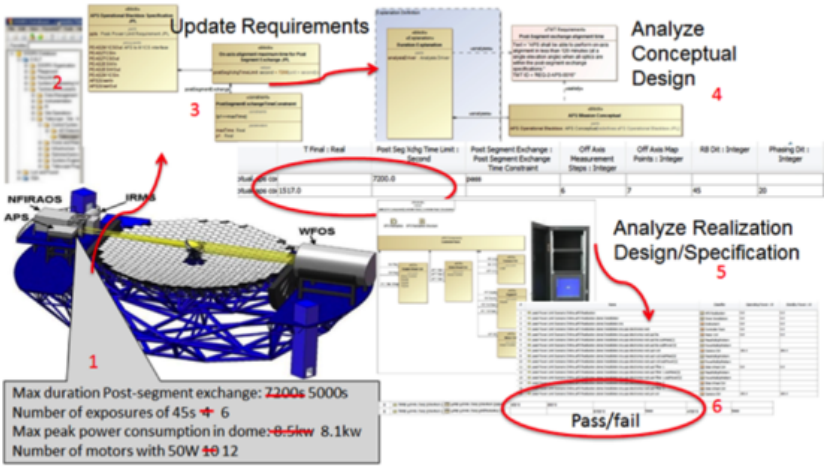


Figure 7. Propagation of a changed requirement. (Used with permission. Permission granted by Jamie Nakawatase.)

In this article, a simplified APS model is shown to illustrate the model-based approach for analyzing changed requirements. The full analysis is available in [4] and [5].

A model-based approach was also applied to APS for error analysis, which was performed to describe the accuracy of expected system performance against requirements. It involved multiple artifacts to analyze a requirement such as “APS shall measure the position of the telescope pupil to an accuracy of 0.03% of the diameter of the pupil.” Defining requirements and parameters in the model indicated the required accuracy and current best estimates of the system design. Defining various roll-up patterns allowed for error decomposition calculations. The benefit is realized in automated requirements verification when applying a parametric solver to formulate results for specified equations in the model. This model-based approach formally integrates accuracy requirements with the system design.

The system model for NFIRAOS LGS MCAO and NGSAO modes was developed to capture sequence behaviors and operational scenarios to run Monte-Carlo simulations for verifying acquisition time, observing efficiency, and operational behavior requirements. The model is particularly useful for investigating the effect of parallelization, identifying interface issues, and re-ordering sequence acquisition tasks.

ESEM enables system analysis by conducting quantitative assessments to select and/or update the most efficient system architecture and generate derived engineering data. System analysis provides rigor for technical decision-making. It includes modeling and simulation, cost, technical risks, and effectiveness analyses, and is used to perform trade studies. In particular, it supports requirements verification, which assesses whether a system design meets its objectives and satisfies the constraints levied by system requirements.

Observed Benefits

The MBSE approach applied to APS and NFIRAOS was motivated by optimization to coordinate the efforts of complex system development. In these applications, implicit dependencies are made explicit in a formal model through the use of ESEM, OpenMBEE, and SysML modeling constructs. Requirements are formalized and tracked directly to the evolving system design. This tight association of requirements within a common environment promotes cross-domain integration and efficient communication among stakeholders. The model is used to automate requirements verification and to generate systems engineering products. The benefit over a traditional document-based approach is that currently disconnected artifacts become related in the model, enabling the production of consistent model-based documentation. Requirements verification is an important analysis conducted in the context of MBSE. To perform this analysis, the requirements, executable behavior, and models predicting the system's performance must be integrated. The ability to integrate these elements using ESEM and the OpenMBEE tooling infrastructure is a significant value proposition for the MBSE approach described in this article. In the formally integrated and executable SysML model, simulations are performed to analyze the impact of changed requirements and verify that requirements are met within specified constraints for various operational scenarios. MBSE enhances information exchange through created visualizations that communicate system behavior. For example, duration analyses were performed to study acquisition time for observing. The use of Monte-Carlo simulations proved how the model-based approach optimized the analysis process. Higher quality analysis results were obtained through the execution of operational scenario runs in an articulated system model, and the model continues to serve as a communication tool across various domains.

Acknowledgements

This research was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA), and at NoMagic. The TMT Project gratefully acknowledges the support of the TMT collaborating institutions. They are the California Institute of Technology, the University of California, the National Astronomical Observatory of Japan, the National Astronomical Observatories of China and their consortium partners, the Department of Science and Technology of India and their supported institutes, and the National Research Council of Canada. This work was supported as well by the Gordon and Betty Moore Foundation, the Canada Foundation for Innovation, the Ontario Ministry of Research and Innovation, the Natural Sciences and Engineering Research Council of Canada, the British Columbia Knowledge Development Fund, the Association of Canadian Universities for Research in Astronomy (ACURA), the Association of Universities for Research in Astronomy (AURA), the U.S. National Science Foundation, the National Institutes of Natural Sciences of Japan, and the Department of Atomic Energy of India.

References

Works Cited

[1] Zwemer, D., “Connecting SysML with PLM/ALM, CAD, Simulation, Requirements, and Project Management Tools”, Intercax LLC, May 2011. [2] Friedenthal S, Moore A., and Steiner R., “A Practical Guide to SysML 3rd Ed.”, Morgan Kaufmann OMG Press, 2014 [3] ISO/IEC, ISO/IEC 42010:2011, Systems and software engineering - Architecture description

Primary References

[4] Karban, R., Jankevičius, N., Elaasar, M. “ESEM: Automated Systems Analysis using Executable SysML Modeling Patterns”, INCOSE International Symposium (IS), Edinburgh, Scotland, 2016 [5] Karban, R. “Using Executable SysML Models to Generate System Engineering Products”, NoMagic World Symposium, 2016 [6] OMG SysML, “Systems Modeling Language (SysML) Version 1.4”, OMG, 2014 [7] Open Source Engineering Environment: https://open-mbee.github.io/

[8] TMT, “Thirty Meter Telescope.” http://www.tmt.org [9] JPL, “Jet Propulsion Laboratory.” https://www.jpl.nasa.gov/ [10] Karban R., Dekens F., Herzig S., Elaasar M, Jankevičius N., “Creating systems engineering products with executable models in a model-based engineering environment”, SPIE, Edinburgh, Scotland, 2016

Additional References

https://github.com/Open-MBEE/TMT-SysML-Model
https://github.com/Open-MBEE/mdk/tree/support/2.5/manual
http://www.omgwiki.org/MBSE/doku.php?id=mbse:telescope
https://groups.google.com/forum/#!forum/openmbee

< Previous Article | Parent Article | Next Article>
SEBoK v. 2.9, released 20 November 2023

Retrieved from "https://sebokwiki.org/w/index.php?title=Applying_a_Model-Based_Approach_to_Support_Requirements_Analysis_on_the_Thirty-Meter_Telescope&oldid=69974"

This page was last edited on 18 November 2023, at 23:16.