

**MBSE IN TELESCOPE MODELING: EUROPEAN EXTREMELY LARGE TELESCOPE – WORLD'S BIGGEST
EYE ON THE SKY
TOOL VENDOR PERSPECTIVE**

Saulius Pavalkis, Ph. D

No Magic, Inc., saulius.pavalkis@nomagic.com

ABSTRACT

The most ambitious project of the European Southern Observatory's (ESO) is the construction of the European Extremely Large Telescope (E-ELT) which will be by far the world's largest optical and near-infrared telescope, and will provide images 15 times sharper than those from the Hubble Space Telescope. Such a project poses continuous challenges to systems engineering due to its complexity in terms of requirements, operational modes, long operational lifetime, interfaces, and number of components. Since 2008, the Telescope Control System (TCS) team has adopted a number of Model Based Systems Engineering (MBSE) practices in order to cope with the various challenges ahead.

This is one of the largest publicly available MBSE information sources. This includes: the complex, interdisciplinary real world sample model¹ recommendations, findings, issues, Open-Source MBSE Plugin for creating the model structure, extracting model variants, and supporting model based document generation based on DocBook, and multiple publications.

The E-ELT is one of the most influential projects for SysML standards and MBSE tools development. One of the project's and the supporting team's goals was feedback for the MBSE tool vendors and OMG SysML standards. This resulted in hundreds of formal requests (hundreds of tickets in the No Magic, Inc., support system, tens of scientific and industrial papers, MBSE guidance, and inputs to SysML standard update) and informal requests, both types of which clarified the standard and significantly moved forward the MBSE support in tools.

In this paper we will overview MBSE application for this project as the core method to manage the complexity. We will identify major MBSE usage aspects. We will answer why and how MBSE was used for telescope modeling.

INTRODUCTION

About MBSE

Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. ²

It is very important to realize that there is more to modeling than just drawing diagrams. The model adds value to system engineering practices because it can be validated, queried, reasoned about, and used to create documentation and operational artifacts like software. ³

MBSE allows integrating and addressing multiple aspects of the system: requirements, functional / behavioral model, performance model, structural / component model, other engineering analysis models (Exhibit 1). ⁴

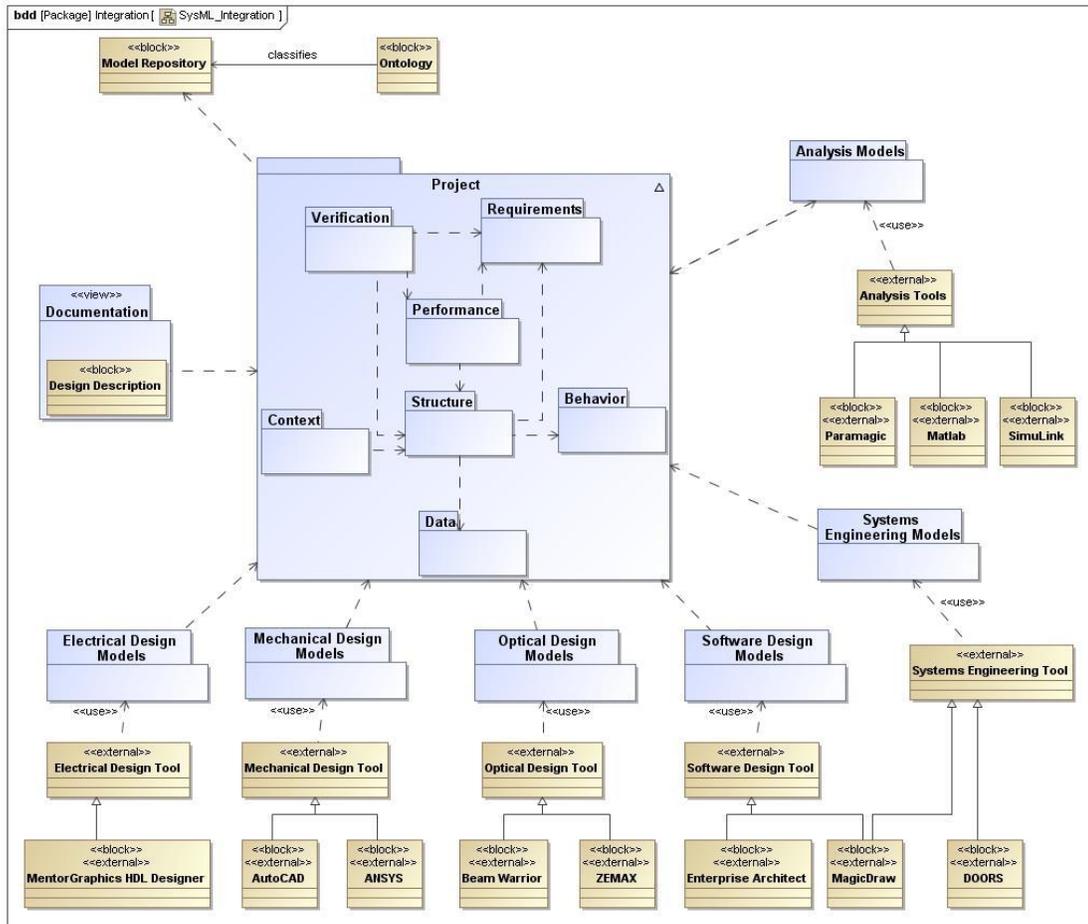


Exhibit 1: MBSE integrating and addressing multiple aspects of the system

Tool Vendor Perspective on MBSE

MBSE is an evolving practice in the early stages of adoption similar to the mechanical and electrical engineering domains in the past century, and the software domain 20 to 30 years ago. Today's growing complexity of systems and the demands of the global marketplace increases the recognition of the potential MBSE in systems engineering.

We, as the highly standard compliant and popular modeling tool vendor, are in the middle of this process: satisfying our industrial client needs on the one hand and influencing and actively developing the system engineering - SysML standards - on the other hand.

3 Pillars of MBSE

MBSE can be decomposed into main 3 aspects:

- **Language** – de facto modeling language is SysML from OMG.
- **Method** – without the method, the language can be used in different ways, including incorrect methods. No Magic, Inc., products are used for projects using different methods: OOSEM, Harmony Process by IBM, SysML Cookbook by SE², and FAS by GfSE. However our tools are independent from these methods.
- **Tool** – a model is collection of complex data structures which can be edited, augmented, queried and reported upon using a suitable tool which is indispensable. No Magic was chosen as the vendor with the

most suitable for such complex projects as telescope modeling. *“The implementation of the SysML specification is a challenge for tool vendors.”*⁶

Two more are identified by telescope modeling team:

- Model Transformation, Model Validation
- Modeling recipes, best practices, and patterns⁷

About the Telescope Project

Challenge team

In the framework of INCOSE’s⁸ strategic initiative, the Systems Engineering Vision 2020, one of the main areas of focus is model-based systems engineering. In keeping with this emphasis, the European Southern Observatory (ESO)⁹ collaborated since 2007 with the German Chapter of INCOSE (GfSE)¹⁰ in the form of the “MBSE Challenge” team SE^2¹¹.

The team’s task is to demonstrate solutions to challenging problems using MBSE.

In 2010, INCOSE presented an award to the challenge team for Achieving the Systems Engineering Vision 2020 for exceptional work and dedication in establishing and managing the Challenge Team in support of the INCOSE mission (Exhibit 2).

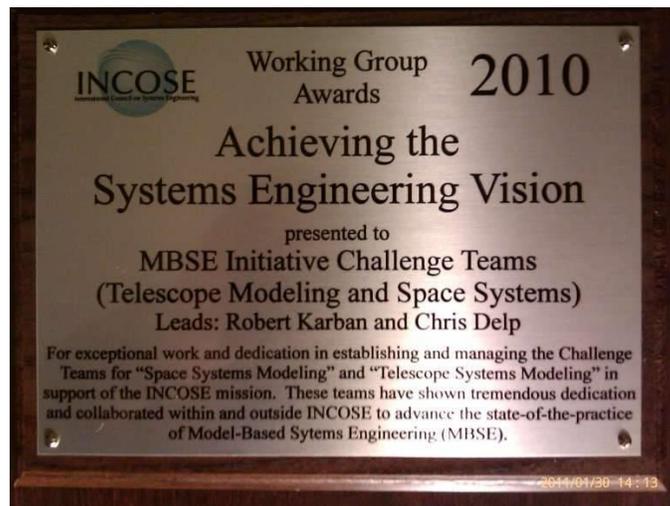


Exhibit 2: INCOSE award

The Active Phasing Experiment (APE) technology demonstrator for the future E-ELT

The Active Phasing Experiment (APE), a European Union Framework Program 6 project, was chosen as the subject of the SE^2 Challenge Team. Many technical products in the telescope domain show an increasing integration of mechanics with electronics, information processing, and optics, and can therefore be considered as an opto-mechatronic system.

The next generation of telescopes needs to collect significantly more light than current telescopes, therefore requiring larger reflecting surfaces that consist of many individual mirror segments. Due to different disturbances (such as vibrations, wind, and gravity), the segments must be actively controlled to provide a continuous mirror surface with a phasing error of only a few nanometers over the main mirror’s diameter of 42 m. The main challenge is to correctly detect the positioning errors of the segments via specific phasing sensors in order to create a continuous mirror surface.

The case study is the Active Phasing Experiment (APE) technology demonstrator for the future E-ELT, which is a high-tech, interdisciplinary opto-mechatronic system in operation at the ESO Paranal Observatory¹². The

essential purpose of the APE experiment is to explore, integrate, and validate active wave front control schemes and different phasing sensor technologies for a European Extremely Large Telescope (E-ELT).

APE was developed to evaluate those sensors, and was installed on one of the 8 m telescopes that constitute part of the Very Large Telescope in Chile (VLT) for sky tests (Exhibit 3, Exhibit 4).¹³

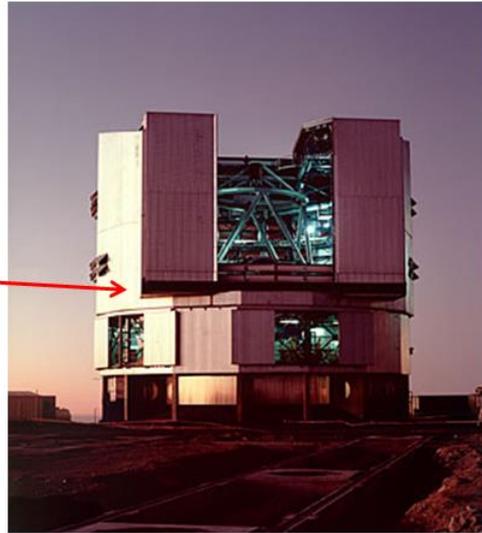
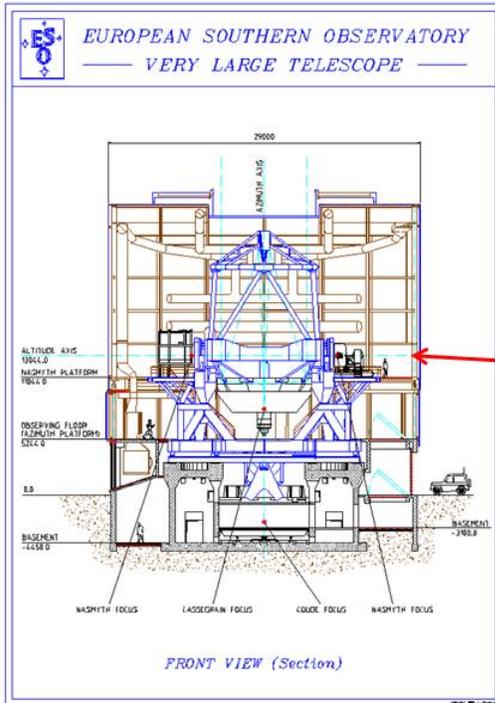


Exhibit 3: APE was installed at telescope in the Atacama desert, Chile

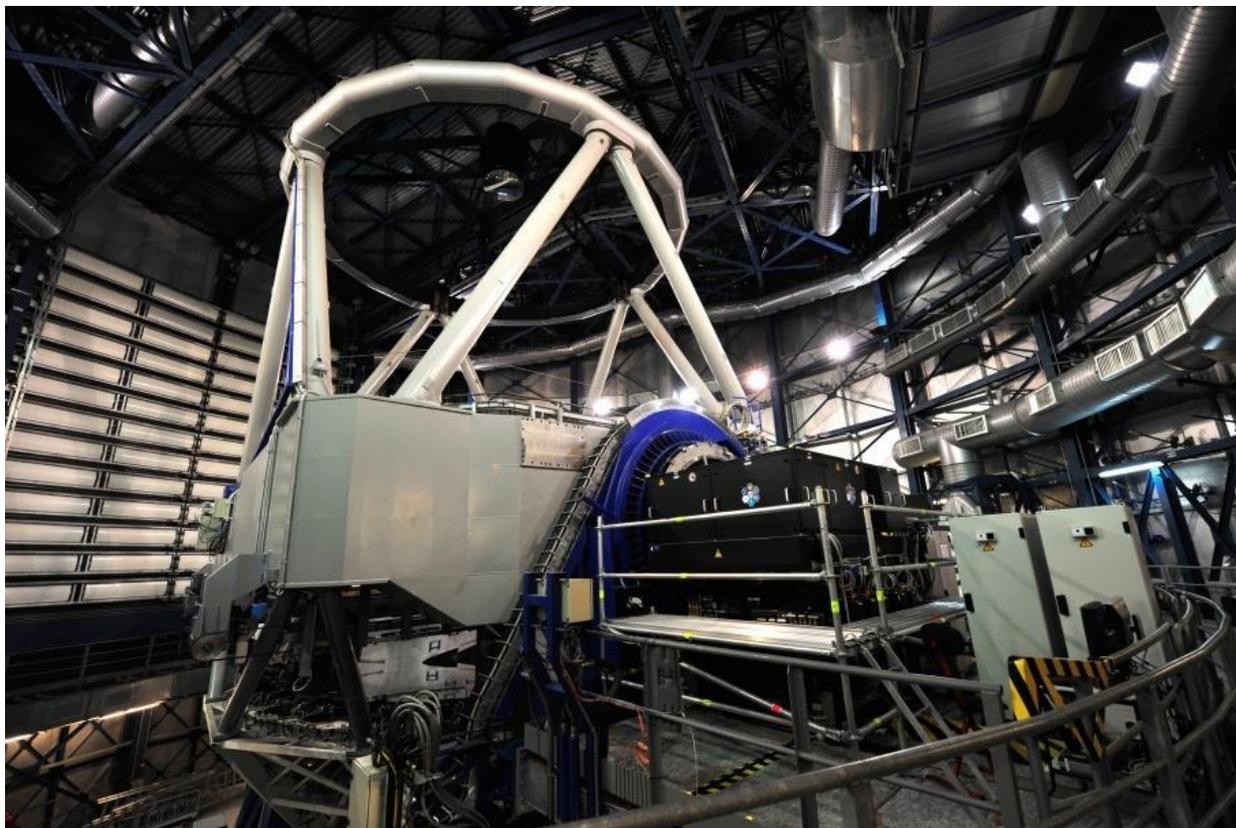


Exhibit 4: APE installed at the telescope

The SysML¹⁴ models were created by reverse engineering from existing documentation and from interviews with systems engineers.

Model for APE project has:

- Three major model parts:
 - Actual system model: APE (with all mentioned system aspects)
 - Catalogue model: standard parts, library of block prototypes
 - Modelling profile: additional stereotypes
- Main characteristics:
 - Scalable model structure and organization
 - Includes model annotations, external references
 - Various examples of ports and flows to model interfaces
- Abstraction levels
 - Functional, Structural, Deployment¹⁵

The experiences of the challenge team gained in the APE project influenced efficient MBSE adoption for the Telescope Control System (TCS).

Telescope Control System (TCS)

MBSE and SysML were chosen to model the Telescope Control System (TCS), which controls many different opto-mechanical parts and devices partially embedded in the telescope structure:

- The Control System includes all hardware, software and communication infrastructure required to control the system
- 10,000 tons of steel and glass
- 20,000 actuators, 1,000 mirrors
- 60,000 I/O points, 700Gflops/s, 17Gbyte/s
- Manages and coordinates system resources (subsystems, sensors, actuators, etc.)

TCS (Exhibit 5) includes all hardware, software and communication infrastructure required to control the telescope (including the dome) down to, but not including, actuators and sensors. Many sub-systems will be contracted and have to be properly integrated after independent development. Therefore, TCS includes the definition of interfaces, requirements, standards for the field electronics, software, and hardware of sub-systems.¹⁶

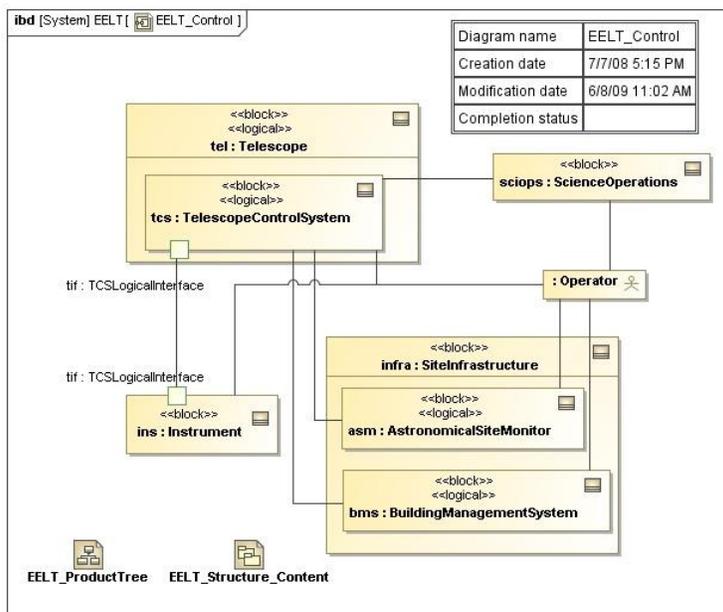


Exhibit 5: TCS Context diagram

The model for Telescope Control System (TCS):

- Defines infrastructure (e.g. network)
- Defines interfaces to sub-systems
- Provides a cost estimate, power consumption
- Defines common standards based on catalogs and design conventions
- Defines requirements for subsystems (e.g. data rates, data volume, latency)
- Maintains a consistent information model of TCS properties to manage its size
- Provides a design which satisfies telescope functions (e.g. wave front control strategies)
- Uses two methods: State Analysis (SA) and the Object Oriented Systems Engineering Method (OOSEM)¹⁷

Why Were MBSE and SysML Used?

The telescope, as an extremely complex system, has a large number of functional, performance, physical and interface requirements which have to be satisfied. The control system alone consists of 12 computing nodes. These elements offer all kinds of optical, mechanical, electronic and software interfaces, with systems internal and external to other systems. Their management alone is very challenging for the systems engineering team, which apply to many complex systems. The team believed MBSE will contribute significantly to manage complexity.¹⁸

The eventual goal is to have a correct-by-construction development process, where a model is used to reason about the proposed solution (formally prove properties of interest through inference e.g. checking design consistency by evaluating the model against constraints), ensuring that all required functionality will be delivered and the correct behavior exhibited, by formally defining and constraining the design space (e.g. like a software engineer bases code development on state machines only). Testing is still performed, but its role is to validate the correct-by-construction process rather than to find bugs.¹⁹

On the other hand, the “MBSE Challenge” team SE² had broader goals:

- Provide examples of:
 - SysML
 - Common modeling challenges
 - Modeling approaches
- Build a comprehensive model which serves as the basis for providing different views to different engineering aspects (e.g. system, logical, mechanical, context) and subsequent activities of analysis and design alike.
- Demonstrate that SysML is an effective means to define common concepts (requirement types, interfaces, relationships, etc.).
- Demonstrate that a SysML model enhances traceability between requirements, design and verification/validation.²⁰

MAJOR MBSE USAGE ASPECTS FOR TELESCOPE MODELING

Next, we overview some of the major areas and core tool capabilities which influenced managing the complexity of this project and MBSE in general.

We organize the sections in following style. We identify used MBSE modeling aspect (project needs), then question why it is important, and how it is achieved. Actual quotes from project team are provided.

We do not intend to prioritize or cover all the aspects, as this project was perhaps the most influential project for standards and tool development. It resulted in hundreds of formal and informal requests which clarified the standard and moved forward MBSE support in tools.

Major MBSE usage aspects:

- Model organization
 - Clear Scalable Project Structure
 - Easy navigation
- Reusability
 - System Context and Clear Interfaces Specification
 - Model Libraries and Profiles
- Model Documentation
- Configuration and Collaboration
- Requirements Specification
 - Model-Based Requirements Specification
 - Integration with Requirements Management Tools
- Different perspectives of the system

- System Structure – Product Tree
- Multiple Internal Structures
- Behavior Model
- Data Model
- Analysis
 - Verification and Validation
 - Variant Modeling and Trade-off Analysis

Clear Scalable Project Structure

Why?

To decrease large system complexity

How?

Project is structured in stable structure which repeats itself from high level to systems structure, to subsystems structure and so on. Each package provides one or more views on specific perspective; e.g. context, structure, data, and so on.

Structure example:

- Objectives and Requirements
- Context
- System Structure
- Behavior
- Data
- Performance
- Verification

Easy navigation

“In general, navigation should happen through the diagrams, and not through the containment tree. Having that in mind, the diagram has a lot more importance, in particular hyper linking diagram elements to other diagrams.”²¹

Why?

Clear understanding how to start and brows the model.

How?

Content diagram is created (Exhibit 6). From top level content diagram it is navigation to subsystem content diagrams and further. Web browser style navigation through the model is an expected navigation method.

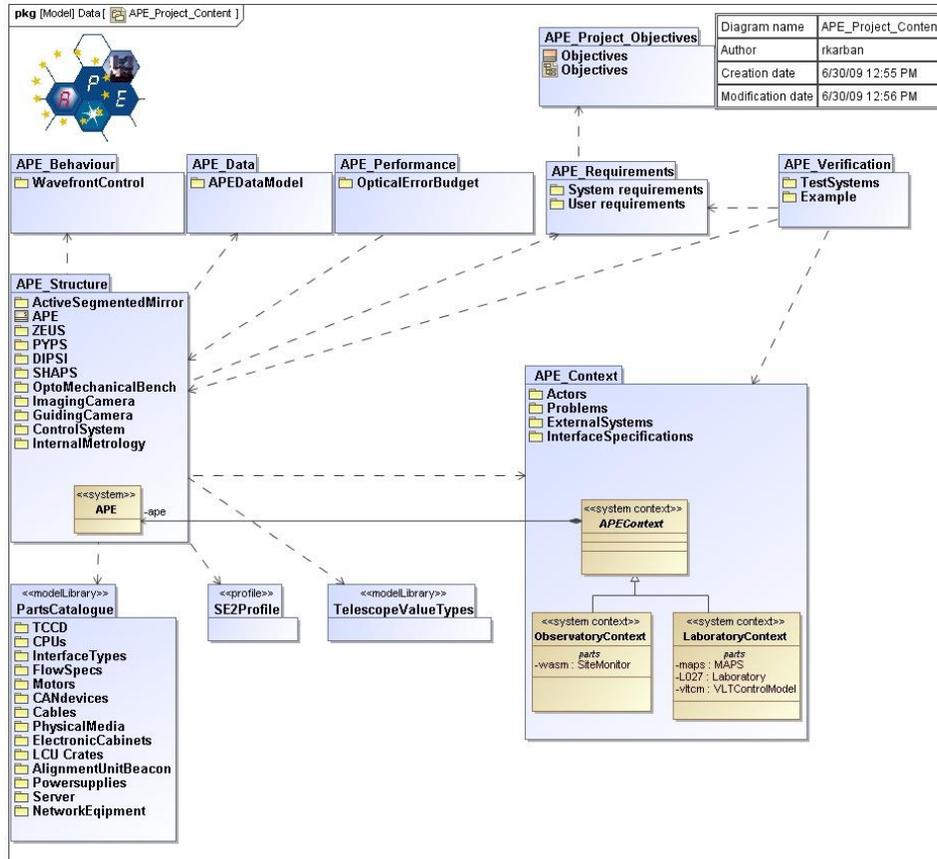


Exhibit 6: Project content diagram

Model-Based Requirements Specification

"APE, as any complex system, has a large number of functional, performance, physical and interface requirements, which have to be satisfied. This implies the need for formal requirements management during the project. APE has about 50 high-level system requirements. The control system has also about 50 requirements, refined by 150 Use Cases."²²

Why?

There is more than one reason:

- Same environment for formal requirements management as for other aspects.
- Traceability between requirements, design, and verification/validation.
- Coverage analysis.

How?

Using OMG SysML standard means for text-based requirements modeling: requirement elements and relations, requirements diagram (Exhibit 7), requirement table (Exhibit 8), and matrix (Exhibit 9). Tool capabilities for model based requirements: documentation generation, integrations and interchange, etc.

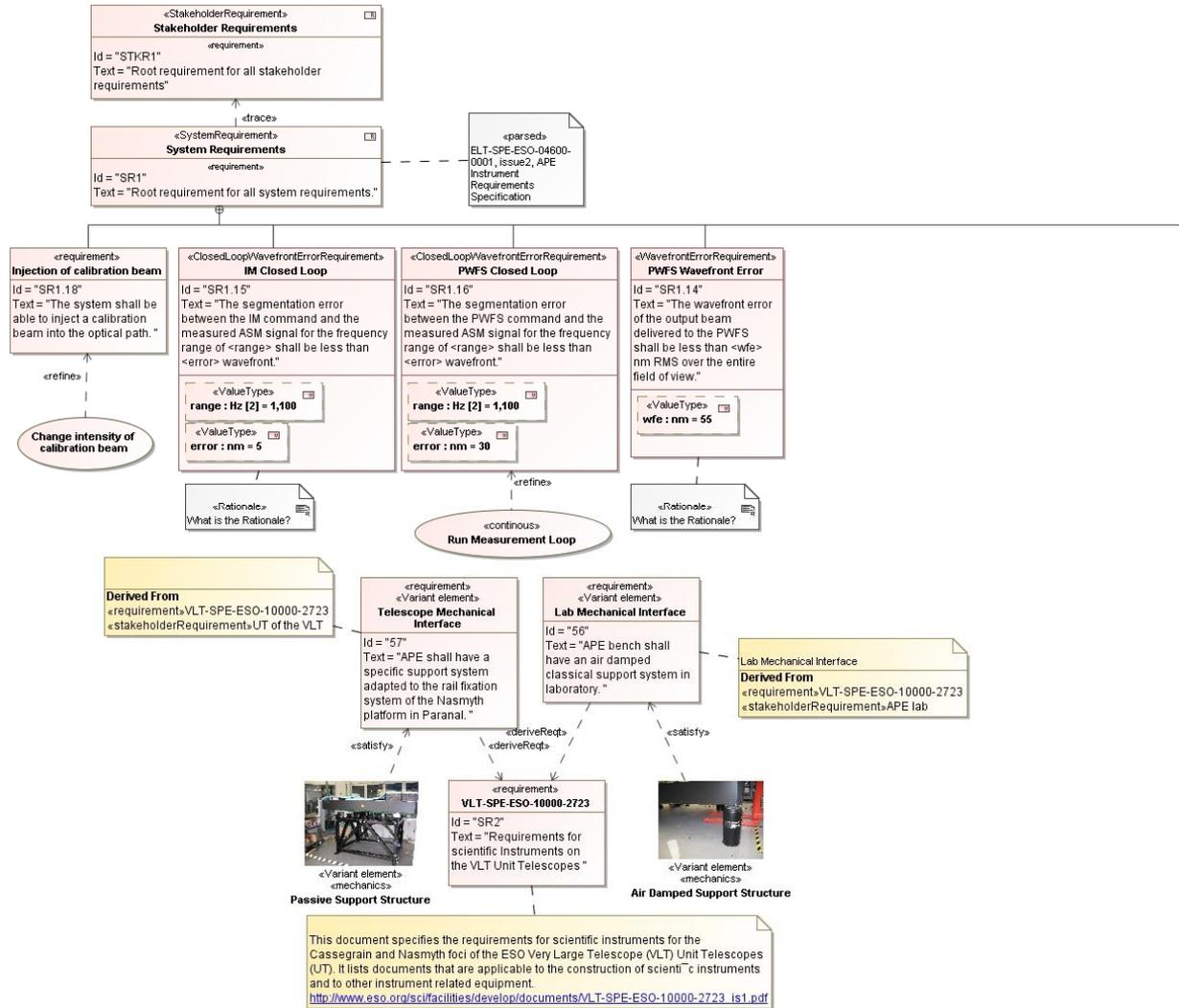


Exhibit 7: Requirements diagram

Tables allow to manage requirements (create, edit, sort, prioritize, filter, publish, import / export) in requirements tools common way. Even requirements are in tables, each line corresponds to a separate model element which can be represented as well in other diagrams, matrix, etc.

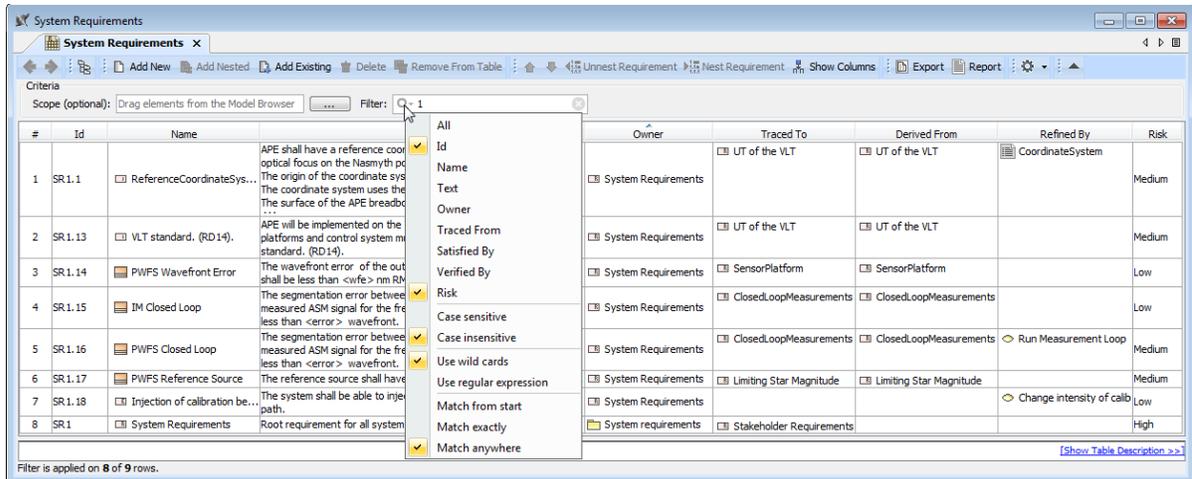


Exhibit 8: Requirements filtering in table

Editable dependency matrix to represent and edit relations.

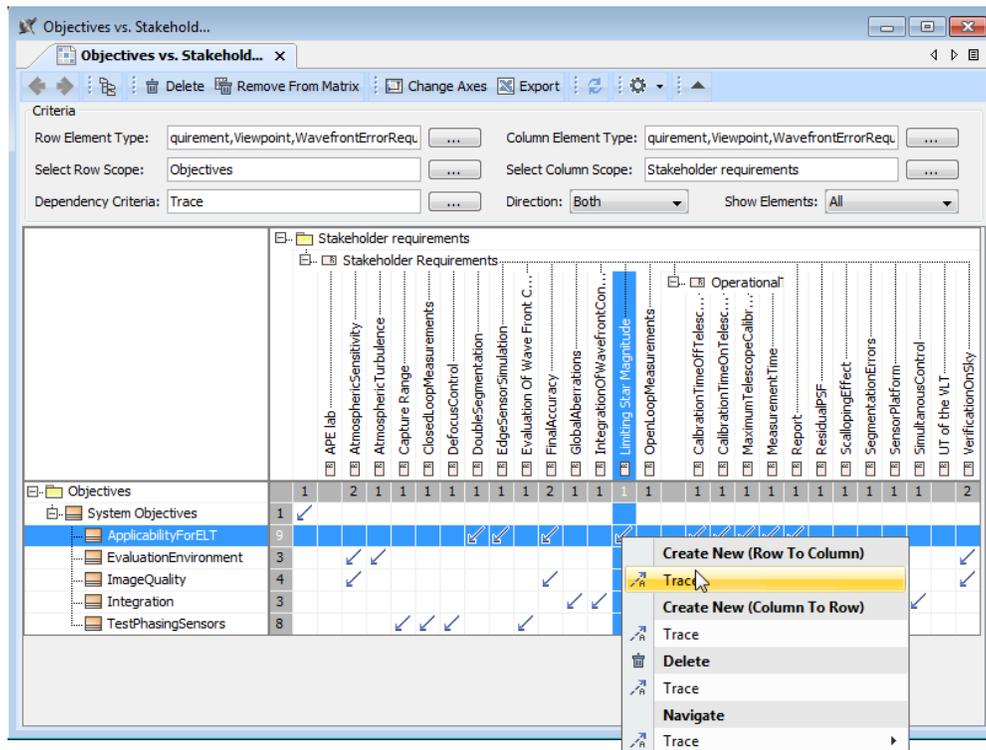


Exhibit 9: Requirements traceability with objectives represented in matrix

Integration with Requirements Management Tools

“structure requirements, in combination with a requirements management tool”²³

“Connecting Requirement Interchange Format (RIF) and SysML offers new SE benefits of requirements tracing and visualization, test case tracing and visualization, incorporating modeled requirements or tests in RM.”²⁴

Why?

Integrations are used to keep synchronization of requirements and design

How?

OMG Requirements Interchange standard based (ReqIF), interchange using dedicated solutions, or 3rd party format such as MS Excel or Comma Separated Values.

System Context and Clear Interfaces Specification

Why?

The system and subsystem context defines the system and subsystems boundaries and allows to specify clear interfaces for reusability. Reusability is key feature as multiple parts are outsourced to contractors or engineering teams.

How?

Modeled using SysML internal block diagrams (IBD). IBD shows a block, its parts and its interfaces (Exhibit 10). The main focus is on system interfaces. The elements available to model interfaces are:

- Ports (Standard UML Ports, Flow Ports)
- Service Interfaces (UML Interface)
- Blocks

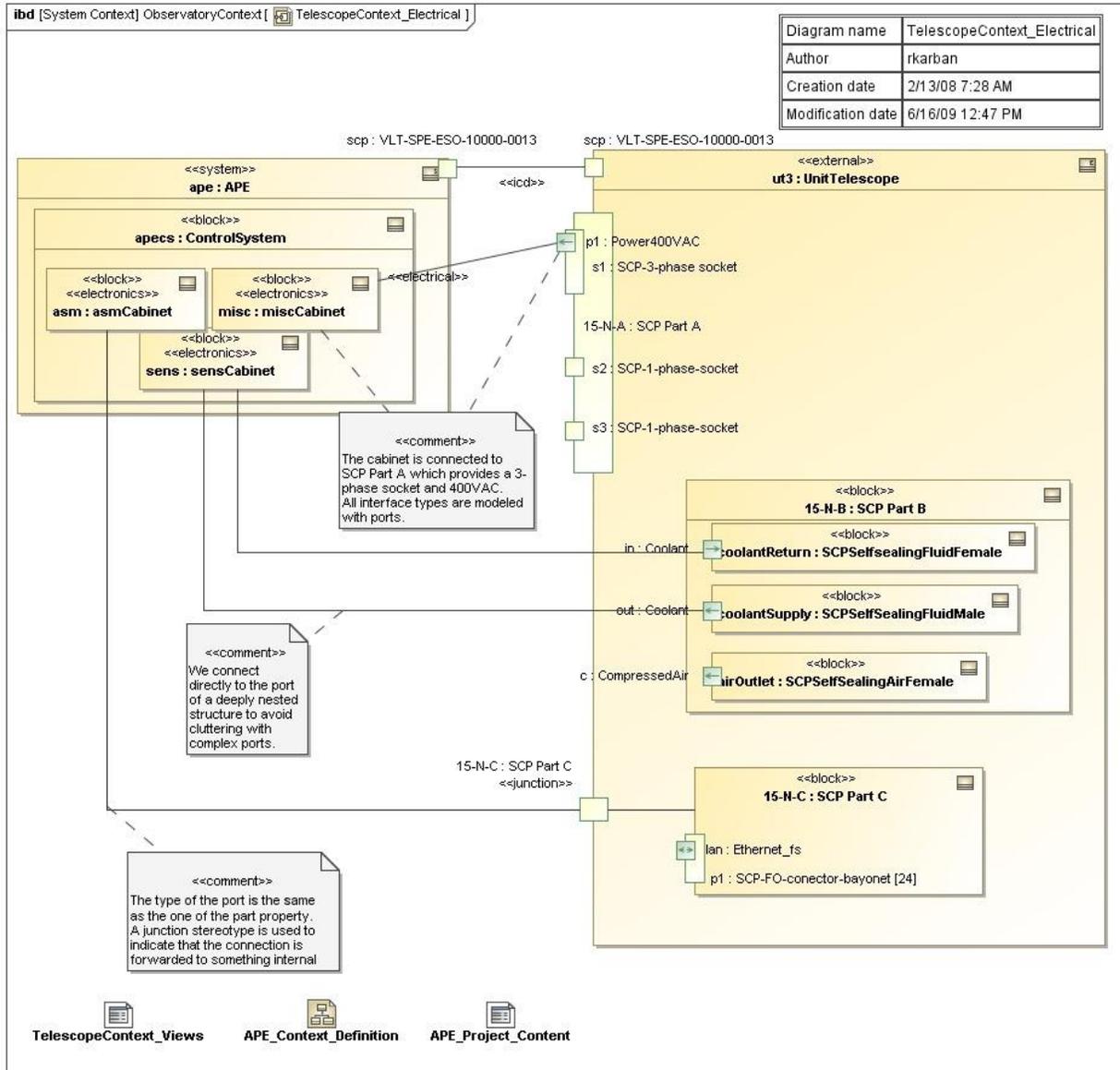


Exhibit 10: Electrical context diagrams

System Structure – Product Tree

Why?

The purpose of the product tree is to give a managerial perspective of which parts need to be built or procured, also cost information for the project.

How?

SysML Block Definition Diagrams (BDDs) are used to model the product tree (Exhibit 11) and SysML Internal Block Diagrams (IBDs) to model different internal structures. The criteria to include elements by means of composition in this diagram is, that they have been built/procured explicitly for the system.

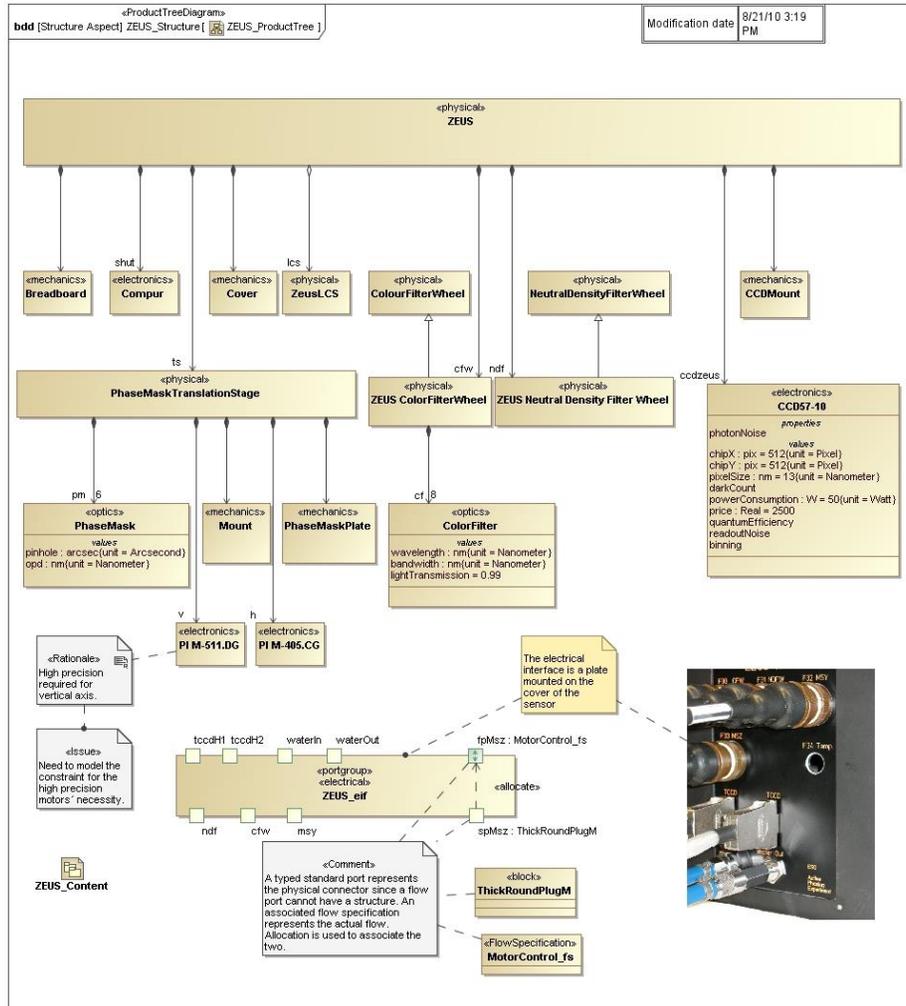


Exhibit 11: Product Tree

Multiple Internal Structures

Why?

Complex system has much more than just one internal structure.

How?

IBD is used to model multiple views showing electrical (Exhibit 10), optical (Exhibit 12), and mechanical elements that are interconnected and therefore multiple structures exist. The same components can be connected in different views in different ways.

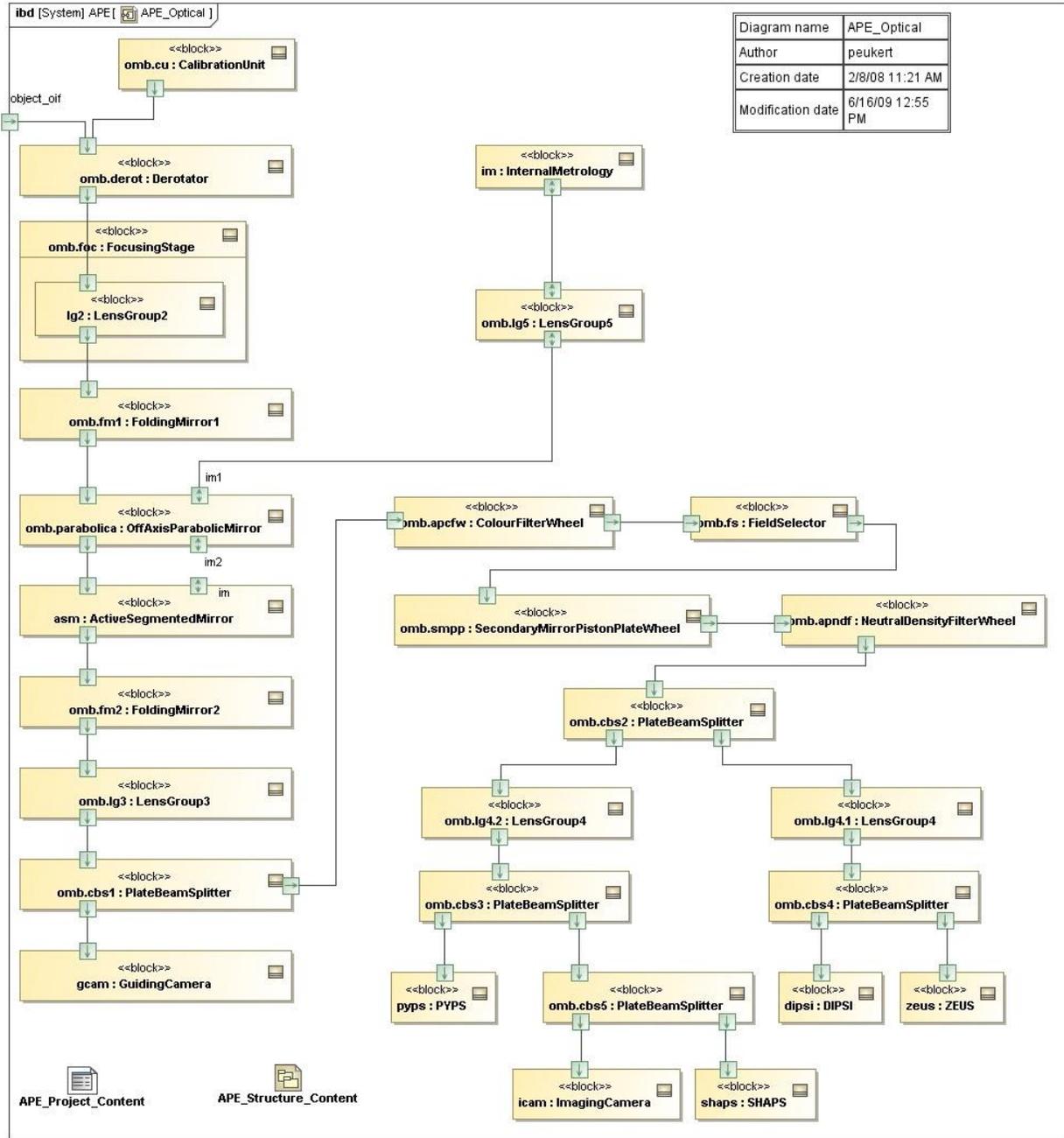


Exhibit 12: Optical Layout

Behavior Model

Why?

To show behavior realized by collaboration of the parts. It is essential to capture the behavior of the system to be able to understand it.

How?

SysML Activity (Exhibit 13), Stata Machine, and Sequence diagrams

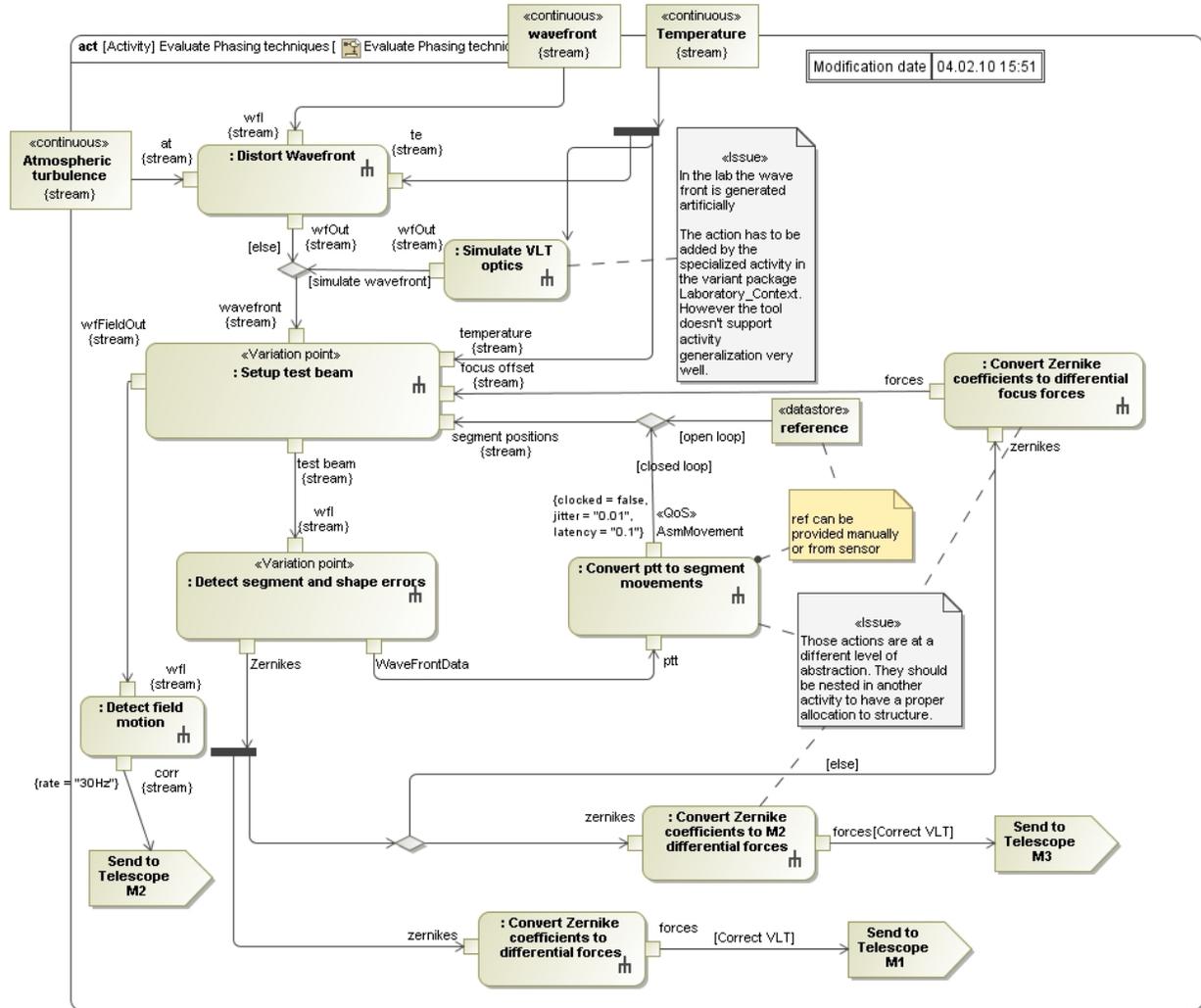


Exhibit 13: Activity for Wavefront control

Data Model

Why?

Shows information/data handled by the system.

How?

SysML provides BDDs for the definition of data (Exhibit 14) and IBDs, activities and sequence diagrams for data usage and flow. The figure shows the composition structure of a measurement and its relation to movements of the segmented mirror (AsmMovement). Those data types are used to define ports in IBDs and objects in Activity diagrams.

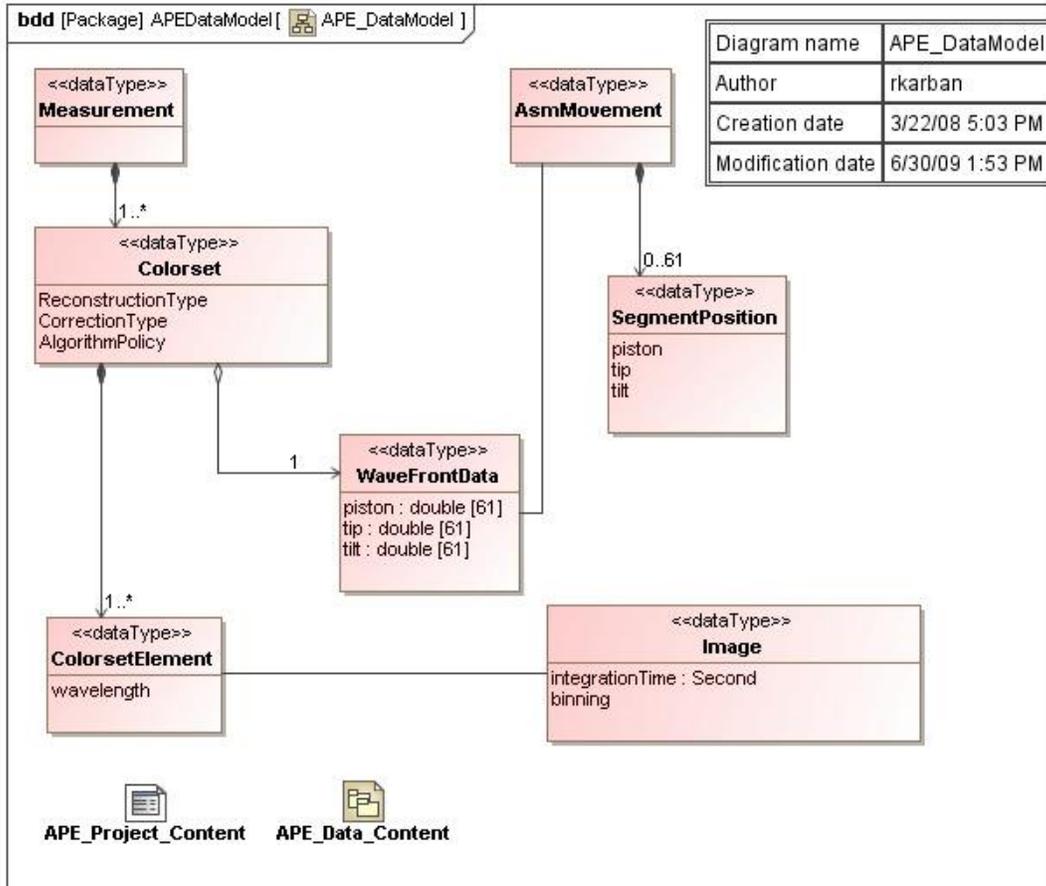


Exhibit 14: BDD for Wavefront data

Verification and Validation

“For every large system, verification is an essential part of the system acceptance in order to prove that the system meets its requirements.”²⁵

Why?

Verification is an essential part of the system acceptance in order to prove that the system meets its requirements. Also it is important to identify project completeness and correctness in early stages as a cost escalation preventative measure. Each stage later where something is discovered adds a significant factor to the cost of fixing it.

How?

SysML supports modelling of TestCases which can be executed using state machine and activity diagrams to verify the future system in early stages. Validation constraints allows the designer or developer to validate the system for completeness and correctness at any point in time.

Model Libraries and Profiles

Why?

Data types, standard parts, library of block prototypes, and other model artifacts that are frequently used are modeled in a model library to increase reuse.

How?

Elements are organized in reusable model libraries, catalogs, profiles which can be used in multiple projects (Exhibit 15). Elements can be easily extended by using inheritance. Custom element types and ontologies are created. This lowers cost for later elements and systems.

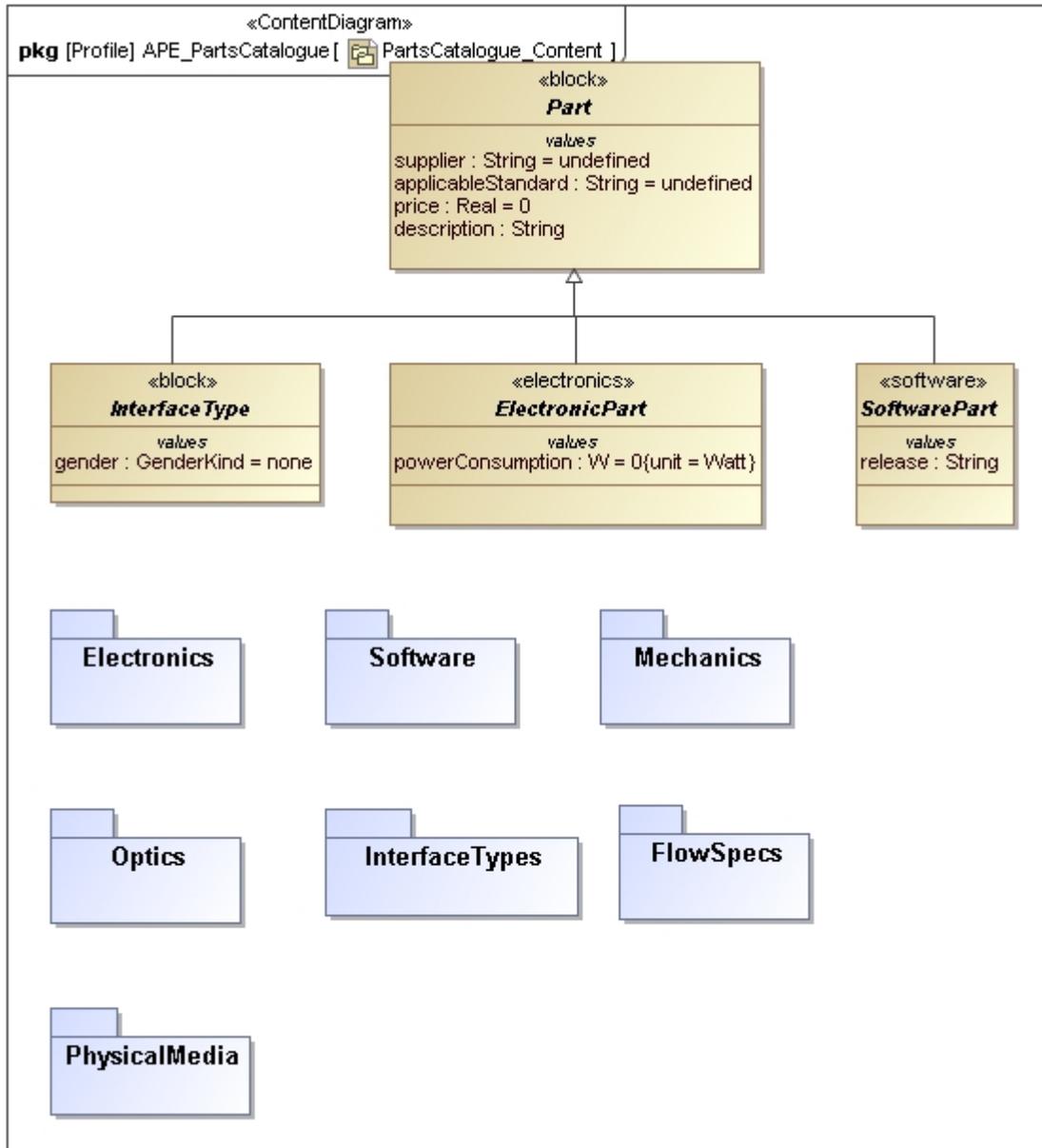


Exhibit 15: Top Level Organization of Parts Catalog

Configuration and Collaboration

*"As soon as a common project model is created and more than one person uses it, configuration control becomes a fundamental requirement... Individual changes must be traceable as well as creating visual differences to follow in detail what has changed where. Due to the extensive linking, side effects (introduced by changes) can go unnoticed and corrupt the model. This can only be mitigated by establishing rigorous configuration management practices and using tools which allow roll-backs."*²⁶

Why?

Enable teamwork and collaboration on single project. Individual changes must be traceable as well as creating visual differences to follow in detail what has changed where.

How?

Using model oriented configuration and collaboration tools, which support all the typical configuration control capabilities, but all of them are tailored for work with models. This includes:

- Accessing and modifying the same model or even the same diagram at the same time with granularity of single element.
- Importing, exporting, updating, committing, merging, comparing, base lining, rollback of changes.
- Splitting projects into subprojects
- Managing access rights to projects.

Model Documentation

*"Diagrams are extracted from the model to create paper-based documentation, as required by the project. The reporting and plug-in facilities of the modeling tool allow creating automatically the recursive structure as defined by the guidelines, cost-estimates using a predefined parts catalogue, and estimates of the required communication infrastructure to accommodate the necessary throughput."*²⁷

Why?

The model shall serve as single source of throughput. Consistency between documentation and model shall be maintained.

How?

Documentation required by official bodies and different stakeholders shall be generated from model. In this way time is saved, and consistency between documentation and model is ensured.

The primary goal of document generation from the model is ensuring that the model will be the source of information and there will be no need to manually create documentation but it will be generated automatically. This is one of the major benefits of MBSE. There are multiple ways how documentation generation is realized; this includes model-based documentation generation based on views and viewpoints which were used in this project.

Variant Modeling and Trade-off Analysis

*"Trade-offs among requirements, system properties, and sub-system properties can easily be evaluated by executing the parametric model of the system. Therefore, requirements can be automatically verified against the design and vice versa."*²⁸

Why?

Used for:

1. Analyzing design alternatives,
2. Evaluating and weight different alternatives via trade-offs, and

3. Modeling of product families.

How?

1. BDD and instance model are used for modeling and analyzing design alternatives (Exhibit 16).
2. Parametric diagram is used for trade-off analysis (Exhibit 17). A parametric relationship states how the value of one property impacts the value of other properties. It is used to support tradeoff analysis by representing evaluation function. The performance indicators, e.g. cost, reliability and performance, are calculated and weighted by an effectiveness function defined as a parametric constraint. The “most effective” variant is the result of the trade-off. Evaluating variants via trade-offs is available by executing parametric diagrams.
3. Dedicated tools are used for modeling of product families.

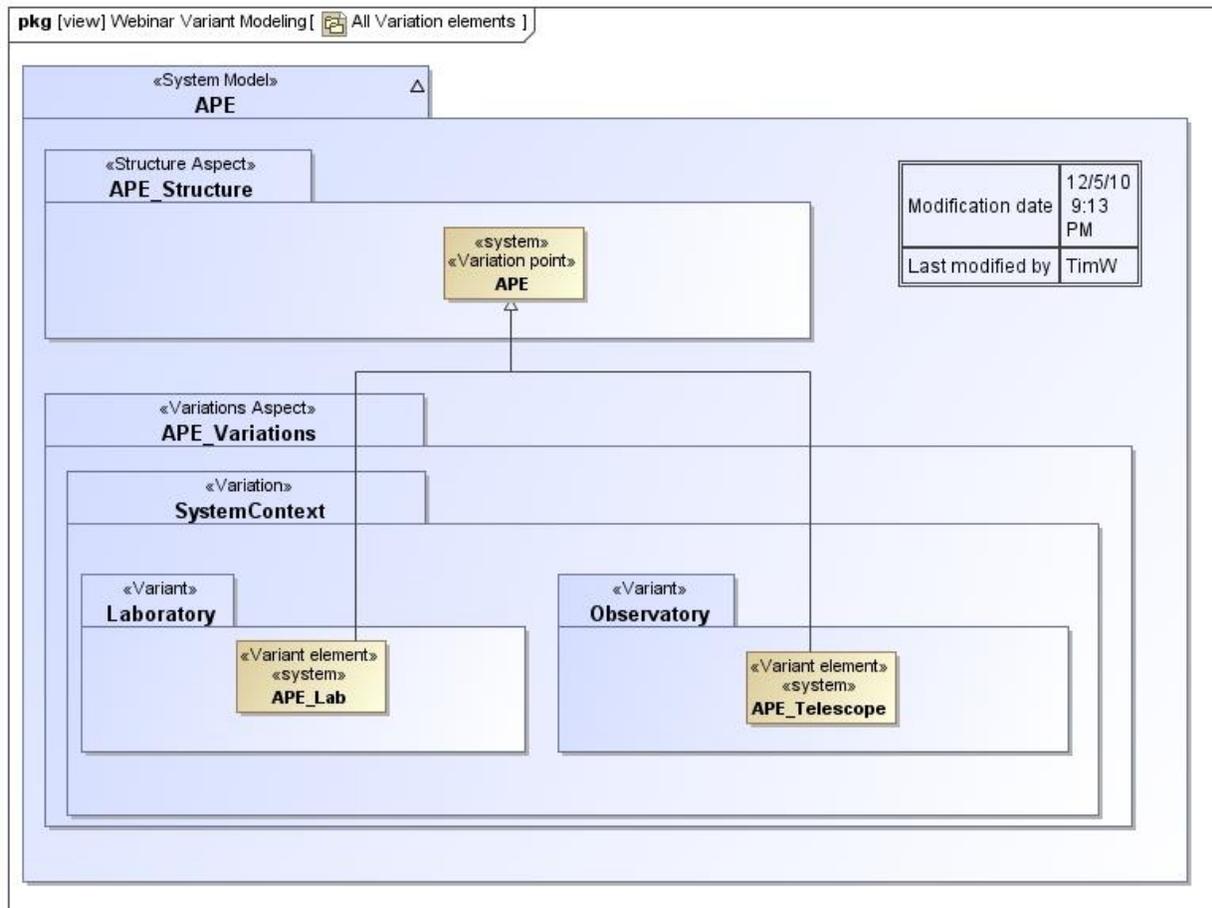


Exhibit 16: Example for variant modeling for analyzing design alternatives

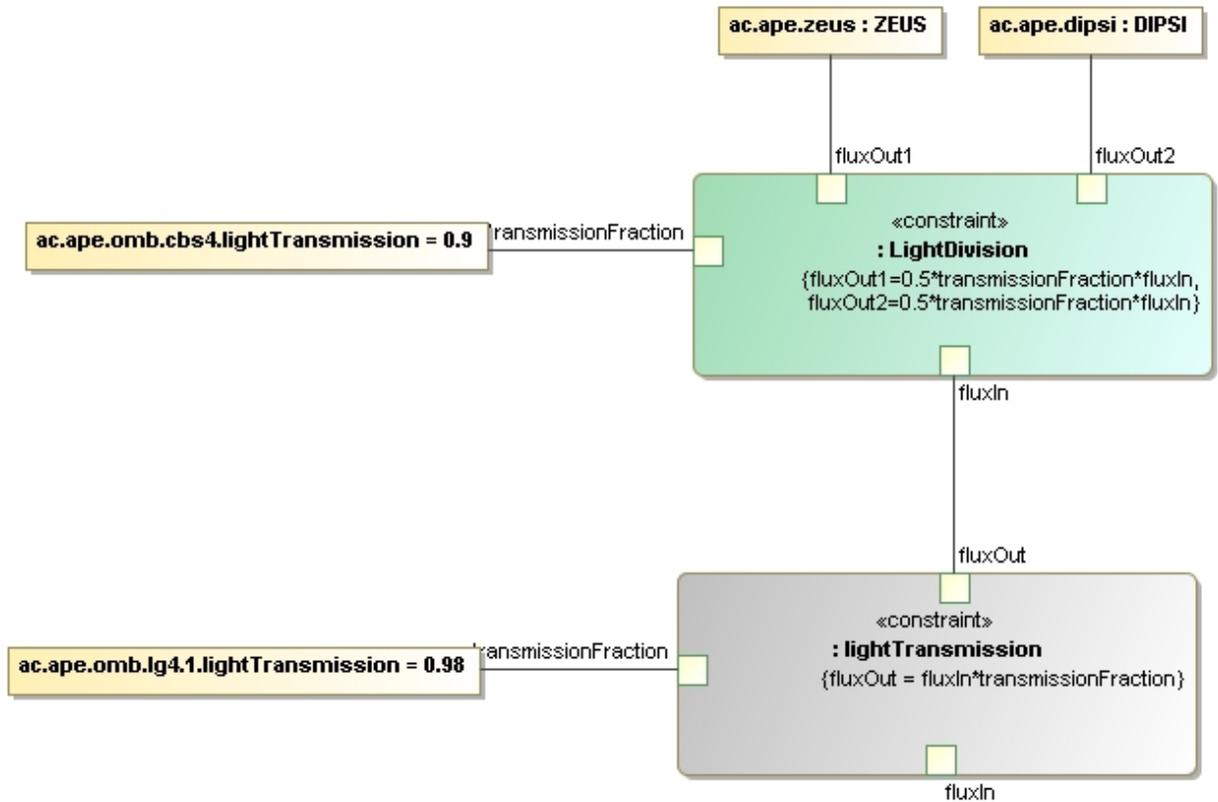


Exhibit 17: Part of a Parametric diagram showing Trade-Off analysis for three different variants.

SUMMARY OF PROJECT RESULTS

In the previous sections we covered major aspect of the project, answered why it is important and how it is achieved. In the following table (Exhibit 18) we summarized results.

Exhibit 18: Project needs coverage with MBSE tools

Area	Project Need	Why?	How?
Requirements specification	Integration with requirements management tools	Transition and synchronization between requirements and design	Standard based interchange through ReqIF Dedicated solution CSV and MS Excel interchange
	Model-based requirements specification	Requirements, design traceability. Coverage analysis	SysML standard and supporting capabilities for model based requirements management
Different perspectives of the system	System structure – Product Tree	Perspective which parts needs to be built or procured	BDD
	Multiple internal structures	Support for complex system structures	IBD
	Behavior model	System behavior part specification	Activity, state machine, and sequence diagrams

	Data model	System data	BDD diagram
Analysis	Variant modeling and Trade-off analysis	Evaluating design alternatives	BDD
		Trade-off analysis	Evaluating variants via trade-offs executing parametric, activity, and state machine diagrams
		Modeling of product families	Dedicated tools
	Verification and validation	Project check for acceptance	executing activity, state machine diagrams, and using validation
Document generation	Document generation	Automatic documentation generation. Consistency between model and specification	Model-based document generation.
			template based document generation
Configuration control & Collaboration	Configuration control & collaboration	Enable teamwork and collaboration on single project	Models oriented configuration and collaboration tools
		Single element level versioning	Next generation database based repository
Reusability	System context and clear interfaces Specification	Clear interfaces for reusability	Interfaces specification
	Model libraries and profiles	Reusability	Profiling, and other extension mechanisms
Model organization	Clear scalable project structure	Decrease complexity	Package structure
	Easy navigation	Model readability	Web browser style project browsing

There were many more challenges and requests identified by the team working on this project; e.g. more than 400 tickets registered in No Magic support system by team members and most of them are fixed or implemented.

Close collaboration with team and high experienced support allowed covering project needs and successful reach other project goals:

- Project has demonstrated that SysML is an effective means to support Systems Engineering
- Model, guidelines and best practices were created in the form of Cookbook for MBSE with SysML
- Gathered experience which was successfully applied in E-ELT and other projects.
- In 2010, INCOSE presented an award to the Telescope Modeling Challenge Team for Achieving the Systems Engineering Vision 2020 for exceptional work and dedication in establishing and managing the Challenge Team in support of the INCOSE mission.
- Impact on other world largest telescope projects, which are also using MBSE, including:
 - The Giant Magellan Telescope
 - JPL NASA Thirty-Meter Telescope
 - The Square Kilometer Array
 - European Southern Observatory's (ESO) projects: VLT, E-ELT, CRRES+

The status of the ELT-TCS MBSE effort can be summarized as follows. The control system team of the E-ELT delivered at the end of 2010, the construction proposal which consists of a complete preliminary design. The tasks

for the successful completion of construction proposal included tasks like defining the control system's infrastructure, power budgets and cost estimates from model based equipment catalogs. The results consisted, among other deliverables, of requirements and ICDs for contracted control systems, an architectural design of the TCS, and design conventions. The goals of enforcing systematic architecture rules correct-by-construction and a consistent verifiable system design have been being gradually achieved by adopting a formal system modeling language (SysML), and fostering the utilization of a common system model.²⁹

Other World Largest Telescope Projects Using MBSE

The Giant Magellan Telescope

The Giant Magellan Telescope (GMT) (Exhibit 19) will be one member of the next class of super giant earth-based telescopes that promises to revolutionize our view and understanding of the universe. It will be constructed in the Las Campanas Observatory in Chile. Commissioning of the telescope is scheduled to begin in 2021.

The GMT has a unique design that offers several advantages. It is a segmented mirror telescope that employs seven of today's largest stiff monolith mirrors as segments. Six off-axis 8.4 meter or 27-foot segments surround a central on-axis segment, forming a single optical surface 24.5 meters, or 80 feet, in diameter with a total collecting area of 368 square meters. The GMT will have a resolving power 10 times greater than the Hubble Space Telescope. The GMT project is the work of a distinguished international consortium of leading universities and science institutions.³⁰

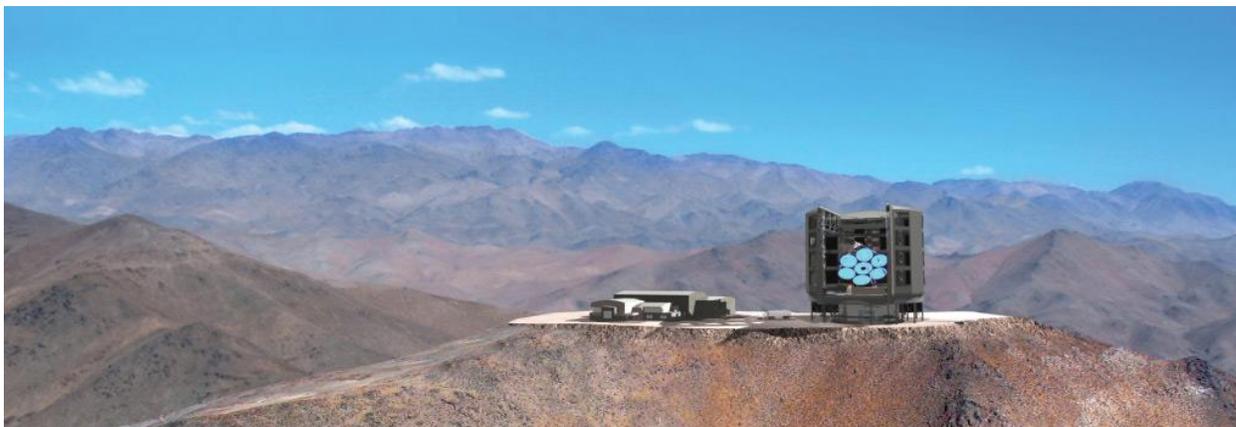


Exhibit 19: The Giant Magellan Telescope

JPL NASA Thirty-Meter Telescope

A 30-meter telescope (Exhibit 20), operating in wavelengths ranging from the ultraviolet to the mid-infrared, is an essential tool to address questions in astronomy ranging from understanding star and planet formation to unraveling the history of galaxies and the development of large-scale structure in the universe.

On May 6, 2014, the TMT International Observatory LLC (TIO) was formed with founding Members from USA, Japan, and China.

The goal of TIO is to design, develop, build and operate the Thirty Meter Telescope.³¹



Exhibit 20: JPL NASA Thirty-Meter Telescope

The Square Kilometre Array (SKA) telescope

SKA (Exhibit 21) is a telescope with up to one square kilometre in collecting surface through an array of antennas distributed over a much larger area. SKA is a radio telescope tens of times more sensitive and hundreds of times faster at mapping the sky than today's best radio astronomy facilities. Simply put: the world's largest radio telescope. Building the SKA will require the development of cutting edge technology and innovation, including the design of the world's fastest supercomputers to process data at rates far greater than the current global internet traffic. The SKA will use thousands of radio antennas, with different antenna technologies. This will enable astronomers to probe the universe in unprecedented detail. The SKA will also be able to survey the entire sky much faster than any radio astronomy facility currently in existence.

Start of construction is scheduled for 2018.³²



Exhibit 21: The Square Kilometre Array (SKA) telescope

European Southern Observatory's (ESO) used MBSE is used for other projects:

ESO adopts MBSE in large scale. MBSE is used for wide spectrum of applications (for example documentation, requirements, analysis, trade studies) and purposes (addressing a particular development need, or accompanying a project throughout many - if not all - its lifecycle phases, fostering reuse and minimizing ambiguity). MBSE usage for telescopes and application area:

- VLT upgrade (Exhibit 22) – specification
- E-ELT (Exhibit 23) - Wave Front Control and Telescope Control System (TCS) specification
- CRIRES+, the CRIRES upgrade - as-is and to-be models specification³³



Exhibit 22: VLT

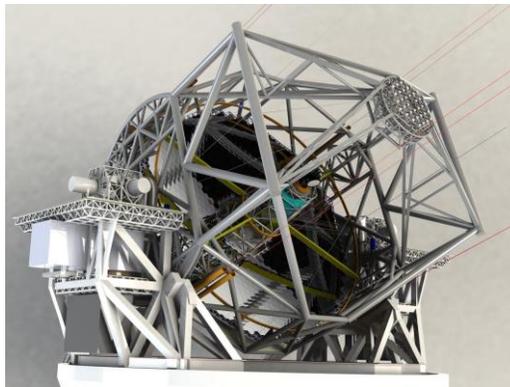


Exhibit 23: E-ELT

CONCLUSIONS

In this paper we overview MBSE application for this project as core method to manage the complexity. We identified the major MBSE usage aspect used for telescope modeling.

We do not intend to prioritize or cover all the aspects, as this is one of the most influential project for standard and tool development. It resulted in hundreds of formal and informal requests which clarified the standard and moved forward MBSE support in tools. E.g. there are more than 400 tickets registered in No Magic, Inc., support system by team members. Most of the reported issues and suggestions are fixed or resulted in new capabilities. Affected capabilities and areas:

- Standard support and usability
- Requirements specification and interchange
- Documentation generation
- Model repository
- Configuration and collaboration
- Activity and state machine simulation
- Validation and verification
- Profiling and extendibility

It is clear that this project did and is still providing significant influence on MBSE Tools development.

From the summary table (Exhibit 18) the major MBSE modeling aspect (project needs) are: clear system model organization, reusability, model documentation generation, configuration and collaboration, model-based requirements specification and interchange, support for different perspectives of the system, verification and validation, variant modeling, and trade-off analysis.

It is important to note that there is more to modeling than just drawing diagrams. The model can be validated, queried, reasoned about, and used to create documentation and operational artifacts like software. All of this is enabled by powerful tools. For an MBSE tool vendor it is a significant challenge to cope with the newest standard updates, and multiple client requests, but at the same time ensure a reliable, fast, and usable environment.

MBSE is not a silver bullet. It pays only on the second or further projects, when previous project parts and experience is re-used, or when changes are required for the project. What significantly helps on the MBSE adoption path is right tools, standards, methods.

MBSE adoption is worth it, but still it is hard work, so choose carefully, to avoid wasting time during the adoption process.

¹ The Active Phasing Experiment (APE) technology demonstrator for the future E-ELT
<http://mbse.gfse.de/extdocs/ape.html>

² INCOSE-TP-2004-004-02, Version 2.03, September 2007

- ³ Karban, R., et al. "Model based systems engineering for astronomical projects." *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2014.
- ⁴ SysML for Telescope System Modeling
http://mbse.gfse.de/documents/GfSE_TdSE2008_MBSETelescopeModeling.pdf
- ⁵ SysML Cookbook by SE^2 <http://mbse.gfse.de/documents/faq.html>
- ⁶ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ⁷ Karban, Robert, et al. "Three years of MBSE for a large scientific programme: Report from the Trenches of Telescope Modelling." *Proceeding 22nd Annual INCOSE International Symposium*. 2012.
- ⁸ INCOSE www.incose.org
- ⁹ ESO <http://www.eso.org/public/>
- ¹⁰ German Chapter of INCOSE (GfSE) <http://www.gfse.de/>
- ¹¹ "MBSE Challenge" team SE^2 <http://www.omgwiki.org/MBSE/doku.php?id=mbse:telescope>
- ¹² ESO Paranal Observatory <http://www.eso.org/public/astronomy/teles-instr/paranal.html>
- ¹³ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ¹⁴ The SysML <http://www.omgsysml.org/>
- ¹⁵ SysML for Telescope System Modeling
http://mbse.gfse.de/documents/GfSE_TdSE2008_MBSETelescopeModeling.pdf
- ¹⁶ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ¹⁷ SysML for Telescope System Modeling
http://mbse.gfse.de/documents/GfSE_TdSE2008_MBSETelescopeModeling.pdf
- ¹⁸ <http://mbse.gfse.de/documents/problem.html>
- ¹⁹ Karban, Robert, et al. "Three years of MBSE for a large scientific programme: Report from the Trenches of Telescope Modelling." *Proceeding 22nd Annual INCOSE International Symposium*. 2012.
- ²⁰ SysML for Telescope System Modeling
http://mbse.gfse.de/documents/GfSE_TdSE2008_MBSETelescopeModeling.pdf
- ²¹ SysML Cookbook by SE^2 <http://mbse.gfse.de/documents/faq.html>
- ²² Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ²³ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ²⁴ SysML for Telescope System Modeling
http://mbse.gfse.de/documents/GfSE_TdSE2008_MBSETelescopeModeling.pdf
- ²⁵ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ²⁶ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ²⁷ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ²⁸ SysML Cookbook by SE^2 <http://mbse.gfse.de/documents/faq.html>
- ²⁹ Karban, Robert, R. Hauber, and T. Weilkiens. "MBSE in Telescope Modeling." *INCOSE INSIGHT 12.4* (2009).
- ³⁰ The Giant Magellan Telescope <http://www.gmto.org/>
- ³¹ JPL NASA Thirty-Meter Telescope <http://www.tmt.org>
- ³² The Square Kilometre Array (SKA) telescope <https://www.skatelescope.org/>
- ³³ Karban, R., et al. "Model based systems engineering for astronomical projects." *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2014.