# Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model – Interim Status #4

David Kaslow[1]
*Consultant, Berwyn, PA 19312 USA*

Bradley J. Ayres[2]
*The Aerospace Corporation, Wright Patterson AFB, OH 45433 USA*

Philip T Cahill
*Consultant, Bryn Mawr, PA 19010 USA*

Chuck Croney
*Consultant, Fort Washington, PA 19034 USA*

Laura Hart
*The MITRE Corporation, McLean, VA 22102 USA*

Alejandro G. Levi[3]
*USAF Space and Missile Center, El Segundo, CA 90245 USA*

**Model-Based Systems Engineering (MBSE) is the formalized application of modeling to support systems engineering. The International Council on Systems Engineering (INCOSE) Space Systems Working Group (SSWG) has been investigating the applicability of MBSE for designing CubeSats by developing a CubeSat Reference Model (CRM). The CRM is intended for instruction and for designing and building a mission-specific CubeSat. Additionally, we are collaborating with Object Management Group (OMG) Space Domain Task Force (SDTF) to develop the CRM as an OMG specification. Our application of MBSE utilizes the Systems Modeling Language (SysML) a graphical modeling language. This paper presents reports on the maturation of the CRM including: 1) CRM as an OMG specification, 2) expansion of architecture and requirement packages to component level, 3) population of architecture and requirement packages using table-based user interfaces, 4) incorporation of technical measures and use cases, and 5) CRM validation strategy.**

## I. Introduction

CubeSats have effectively taken over the university-class launchspace [1]. In 2013, approximately 75% of the university-class missions were CubeSats [2]. When launch failures are factored out, the failure rate of university missions approaches 50% [2]. The design effort for university CubeSats has largely been based on intuition [3]. The rapid growth of CubeSat missions combined with historically high failure rates indicates a need for rigorous systems engineering practices to be applied to university CubeSat missions.

To this end, the International Council on Systems Engineering (INCOSE) Space Systems Working Group (SSWG) began investigating the applicability of MBSE for designing CubeSats in 2011. The initial objective was to demonstrate MBSE as applied to a CubeSat mission

The first phase consisted of developing a SysML model of a CubeSat, and then applying it to the Radio Aurora Explorer (RAX) CubeSat [4]. RAX was a Michigan Exploration Lab and SRI International mission. The purpose was to study the formation of magnetic field-aligned electron density irregularities in the Earth's ionosphere.

The second phase focused on expanding the RAX CubeSat model to include modeling behaviors [3]. This phase of the modeling supported analysis of communication download, power, and mission activities and states and how they were related to opportunities to perform spacecraft functions.

The third phase focused on capturing additional RAX design and operational characteristics [5]. A new RAX CubeSat model was developed. Behavioral models were implemented and tightly integrated with parametrics modeling and analysis to demonstrate trade studies within a systems modeling development environment: 1) No

[1] INCOSE SSWG Chair and AIAA Senior Member
[2] Visiting Assistant Professor of Systems Engineering, AFIT/ENY, Department of Aeronautics and Astronautics, and AIAA Member
[3] AIAA Senior Member

Magic's MagicDraw/Cameo Simulation Toolkit, 2) Phoenix Integration's ModelCenter and MBSE Analyzer, 3) Analytical Graphics Inc.'s Systems Tool Kit, and 4) MathWorks' MATLAB.

The following trade studies were carried out: 1) different configurations of solar panel areas and maximum battery capacity were traded off to compare on-board energy levels and 2) different orbital altitudes and ground station network configurations were traded off to compare quantity of data downloaded.

These three phases fulfilled the initial INCOSE objective. Because the initial model was RAX-centric and not suitable for reuse, two additional objectives were added to catalyze the use of a standard model by the CubeSat community:

- Develop a CubeSat Reference Model (CRM) that a CubeSat team can use as a starting point for its mission
- Develop the CRM as an Object Management Group (OMG) specification

References [6-11] address the additional objectives. The inclusion of behaviors is addressed in Reference [9] and the inclusion of technical measures is addressed in Reference [10]. The CRM validation strategy is outlined in Reference [11].

This paper provides a current status of the SSWG's effort to develop and standardize a CRM. It is organized into six principal sections. Section II provides background information on CubeSats, MBSE, and SysML; Section III describes the CubeSat Reference Model in some detail and includes several diagrams output by No Magic's Cameo System Modeler (CSM); Section IV discusses the process by which CRM is being established as an OMG specification; Section V discusses the implementation of CRM in CSM, and the process by which it will be used to create a mission-specific model; Section VI provides a detailed description of how the CSM implementation will be validated; and Section VII discusses the Next Steps the SSWG is planning to develop further the CRM in CSM.

## II. Background

### A. CubeSat

CubeSat, a type of nanosatellite, is a low-cost standardized satellite with its origin in the CubeSat Project which was established in 1999 by California Polytechnic State University (Cal Poly), San Luis Obispo and Stanford University's Space and Systems Development Laboratory (SSDL).

The basic CubeSat unit is 10x10x10 centimeters with a mass of about 1.3 kilograms, and this cubic unit is referred to as 1U. Over the years, this basic form factor has been modified to include larger form factors such as 3U, 6U and 12U CubeSats. They are typically launched as secondary payloads or deployed from the International Space Station.

### B. 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. References [12] and [13] provide additional information about MBSE.

In MBSE, requirements and design are contained in the model rather than in a series of independent engineering artifacts. Our application of MBSE uses Systems Modeling Language (SysML) and a graphical modeling tool from No Magic.

### C. SysML

Systems Modeling Language (SysML) is a graphical modeling language for modeling complex systems including hardware, software, information, personnel, procedures, and facilities [13].

Utilizing the sematics of SysML, packages are used to organize model elements. A block is the fundamental SysML modeling element. A block can define 1) a type of logical or conceptual entity; 2) a physical entity, a hardware, software, or data component; 3) a person, a facility, an entity that flows through the system; or 4) an entity in the natural environment.

SysML has modeling elements for: requirements, structures, behaviors, and parametrics. Structural elements and their relationships are presented in block definition diagrams and internal block diagrams. Behaviors describe how a block deals with inputs and outputs and changes to its internal state - what the system must do to meet requirements. Behaviors are presented in activity, state machine, sequence, and use case diagrams. Parametrics are the mathematical formulations needed by the models and simulation.

## III. CubeSat Reference Model

The CubeSat Reference Model (CRM) adheres to MBSE, SysML, and primary references including the INCOSE Systems Engineering Handbook [12], A Practical Guide to SysML [13], NASA Systems Engineering Handbook [14], and Space Mission Engineering: The New SMAD [15].

The CRM is systems engineering methodology-agnostic. The mission-specific team can import the CRM into its graphical modeling tool to initiate its process for architecting, designing, and developing its mission-specific Cubesat model. This model will be a repository for the systems engineering artifacts created by the mission-specific team.

Each subsection below addresses an element of the CRM which the SSWG has found to be of interest to practitioners and important for the application of systems engineering to CubeSat development efforts.

### A. CubeSat Reference Information

The SSWG has created a CubeSat Reference Information repositry to accompany the CRM. It is the repository for definitions and for references, and for general, SysML, CubeSat subsystems, and Ground subsystems terminology. The modeling tool underlines any terminology with a definition provided in the CubeSat Reference Information repository. Hovering over the terminology reveals the definition.

### B. Stakeholders

Licenses and regulations, timelines, and procedures must be well understood and included in the CRM. In the U.S. the FCC regulates the radio frequencies, NASA provides orbital debris guidelines, and NOAA regulates remote sensing. The CRM includes stakeholder model elements that represent regulatory agencies. In addition, mission-specific stakeholders are also included such as : sponsor, user, operator, project engineer, mission engineer, developer, integrator, and tester.

### C. Stakeholder Concerns

A stakeholder concern can be manifest in many forms, such as in relation to one or more stakeholder needs, goals, expectations, responsibilities, requirements, design constraints, assumptions, dependencies, quality attributes, architecture decisions, risks or other issues pertaining to the system.

There can be a number of stakeholders and a number of concerns. The needs, objectives, constraints, and requirements result from review, assessment, and integration of the varied concerns.

These model elements and their relationships are mission- and engineering methodology- specific. The mission and methodology may dictate starting with stakeholder concerns, or starting with mission objectives and mission constraints; or there may be a simpler approach of starting with just mission requirements.

Figure 1 shows the relationships that can be established between stakeholder concerns, technical measures, use cases, and requirements. The CRM provides a library of model elements but does not dictate what model elements and relationships to use.

### D. Requirement Packages

The CRM requirements package hierarchy is shown in Fig. 2 and has been expanded to include CubeSat and Ground subsystems component packages.

Each requirements package has a table containing a "header' requirement element that establishes a prefix and numbering scheme which can be changed easily. These tables provide the capability to add, delete, and modify requirements. The requirement tables can be exported to spreadsheets.

The requirement elements and tables can expose requirement properties including: id, name, text, traced to, derived from, refined by, verify method, verified by, and satisfied by. The exposed properties can easily be added to or hidden. Refer to Fig. 3 Requirements Properties.

These properties are defined in the CubeSat Reference Information. For example, the satisfy relationship is used to assert that a model element corresponding to the design or implementation satisfies a particular requirement. The relationship is shown with a dashed line pointing to the requirement. Proof that the assertion is true is accomplished by the verify relationship. Refer to Fig. 3 for an illustration of the satisfy relationship.

The CRM contains requirements stereotypes as shown in Fig. 4. They can be modified to include additional properties as needed.

### E. Use Cases

Use cases are used to refine requirements. The CRM has use case packages at the Enterprise, CubeSat, and CubeSat subsystems hierarchy level. Refer to Fig. 5 Use Cases Population and Fig. 6 Enterprise Use Cases Population.

A use case can be described in a Use Case Description document as part the CRM use case package and then, if warranted, detailed in a SysML model. The document can be simple narrative or have a more detailed format: use case name, scope, primary actor, supporting actor, stakeholder, precondition, trigger, postconditions [9].

A use case package contains use case SysML elements, Use Case Description documents, a table of use cases elements, and Use Cases Population Instructions document.

The Use Case Table exposes UseCase SysML element properties: name, documentation, refines, and active hyperlink (to the Use Case Description document). The Use Case Description and Population Instructions documents reside within the CRM.

### F. Architecure Packages

The CRM provides a CubeSat logical architecture. The logical components are abstractions of the physical components that perform the system functionality without imposing implementation constraints. The physical architecture defines physical components of the system including hardware, software, persistent data, and operational procedures. The logical components are a starting point for a mission-specific CubeSat logical architecture, followed by the physical architecture and the CubeSat development.

The CRM architecture package hierarchy is shown in Fig. 7 CubeSat Domain, Fig. 8 CubeSat Enterprise, Fig. 9 CubeSat Subsystems, and Fig. 10 Ground Subsystems.

The CubeSat and Ground subsystems packages contains subsystems population packages as shown in Fig. 11 CubeSat Subsystems Population and in Fig. 12 Ground Subsystems Population. These packages provide the capability to add, delete, and modify subsystems. Subsystems (e.g., Propulsion) can be eliminated if not needed. Subsystems (e.g., Attitude Determination and Control subsystem and Guidance, Navigation, and Control) can also be combined.

The CubeSat and Ground subsystems are generically and loosely defined as a starting point for the mission-specific CubeSat team. The CubeSat team determines what subsystem capabilities are needed and whether they are provided by software, hardware, persistent data, or operator procedures.

The CRM architecture hierarchy has been expanded to include CubeSat and Ground subsystems components packages. The CubeSat component packages have been populated with representative subsystem components. For example, see Fig. 13 CubeSat Mission Payload Representative Components and Fig. 14 CubeSat Attitude Determination and Control Representative Components.

The CubeSat Subsystems Component packages contain component population packages. Examples are shown in Fig. 15 CubeSat Mission Payload Components Population and in Fig.16 CubeSat Attitude Determination and Control Subsystem Components Population.

The mission-specific CubeSat team populates the CubeSat and Ground component packages with their mission-specific components.

### G. Technical Measures

Technical Measures (TMs) are an established set of measures based on the expectations and requirements that are tracked and assessed to determine overall system or product effectiveness and customer satisfaction. Common terms for these measures are Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and Technical Performance Measure (TPMs).

TMs provides a stakeholder insight into the definition and development of a technical solution. Verification activities provide data to the technical measurement process that are used to assess how well the TMs are either projected to meet, or are meeting, their stated value.

Technical Measure Specifications capture descriptions of technical measures in textual form. Stakeholders will likely describe their TMs as text. They will communicate and negotiate these descriptions with engineers who will transform them into measures that can be tracked and assessed.

MOEs are operational measures of success that closely relate to the achievement of mission objectives being evaluated, in the intended operational environment under a specified set of conditions.

MOPs are measures that characterize the physical or functional attributes relating to the system operation; i.e., they provide insight into the performance of the specific system.

TPMs are a measure of the attributes of system elements used to determine how well the system element is satisfying, or expected to satisfy, specified technical requirements.

An approach has been developed for incorporating TMs in the CRM that begins with a set of mission objectives derived from stakeholder concerns [10]. The objectives are represented as elements captured in the system model. Next, MOEs are derived from the mission objectives. Initially, these MOEs are captured in a special model element that allows for the MOEs to be described in a natural language format that stakeholders are able to understand. Those initial MOEs are then quantified and captured as value properties of the Enterprise block. The MOEs are traced back to their originating source in the mission objectives. Refer to Fig. 17 MOE Specification Population.

MOPs are derived from the enterprise-level MOEs and captured as value properties of the System block. The derivation of the MOPs is captured through the development of constraint blocks and parametric diagrams. This provides for traceability between MOPs and MOEs and supports performance analysis of the MOPs to predict if the MOEs will be met. MOPs are also traced to system requirements captured in the system model. Refer to Fig. 18 MOP Specification Population.

The process steps at the system-level are repeated at the subsystem-level to derive TPMs. These TPMs are traced back to MOPs and subsystem requirements in the same manner as described for MOPs. Refer to Fig. 19 TMP Specification Population.

The Technical Measures Population packages contain technical measures methods. A tchnical measures Word document or Excel spreadsheet contains the methods to calculate technical measures. These methods can be implemented and executed in a constraint block constraints and parameters compartments within a parametric diagram. Of course, these methods can be implemented in a spread sheet. The document or spreadsheet is part of the model and can be edited within the model

Refer to Fig. 20 for the technical measures stereotypes.

## IV. CRM as an OMG Specification

OMG has initiated a process to establish and provide a CRM as a normative OMG specification. It must be SysML compliant and implementation free. It is a template model that provides building blocks that can be specialized to support MBSE CubeSat design.

This paper describes CRM as implemented in No Magic's Cameo Systems Modeler (CSM). But an implementation of the CRM cannot be an OMG specification. However, an XML Metadata Interchange file (XMI) file, which is implementation free, can be extracted from the CSM CRM and submitted to OMG in accordance with its processes for soliciting, assessing, and selecting a CRM. The XMI file will be an OMG normative specification. The submission to OMG will include as non-normative material: 1) a CSM implementation of the CRM and 2) a CSM web published HTML file that can be imported into Internet Explorer.

A mission-specific CubeSat team will download the CRM Specification XMI file from OMG and import it into its own graphical modeling tool (e.g., CSM or Sparx Systems' Enterprise Architect). It is anticipated that the mission-specific CubeSat team will use non-normative material to guide development of their mission-specific CubeSat model.

## V. CSM Implementation of the CRM

A mission-specific team determines, the level of architecture that they start at, depending on their level of experience with space-ground systems, MBSE, and SysML:

- Fundamental with higher level packages
- Representative with additional packages for subsystems as defined in the CSM CRM
- Detailed with additional packages for subsystem components as defined in the CSM CRM.

Table-based user interfaces exist for modifying, adding, deleting design information:

- Mission: Stakeholders, stakeholder concerns, needs, objectives, constraints, requirements
- Specifications: MOE, MOP, and TPM technical measures
- Requirements: CubeSat, CubeSat Subsystems, CubeSat Subsystem Components, Ground Segment, Ground Subsystems, Ground Subsystems Components
- Structures: CubeSat Subsystems, CubeSat Subsystems Components, Ground Subsystems
- Use Cases: Enterprise, CubeSat, CubeSat Subsystems

The development of a mission-specific model includes the following steps:

- Mission-specific enterprise needs, objectives, constraints, and measures of effectiveness are populated
- Mission use cases are created to refine mission requirements which support the measures of effectiveness, objectives, needs, and constraints

- Space and ground segment requirements are derived from mission requirements
- Segment use cases are created to refine segment requirements which in turn trace to mission use cases.
- Segment measures of performance are created in support of the enterprise measures of effectiveness
- Segment requirements are created in support of the measures of performance.
- Space and ground subsystem requirements are derived from segment requirements and trace to segment use cases
- Subsystem technical performance measures are created in support of segment measures of performance and enterprise measures of effectiveness
- Although the CRM Space and Ground Subsystems have been broadly defined, the mission teams may find it necessary to modify the subsystem definitions according to the allocated requirements.

The physical architecture is created from the logical architecture by determining the types of subsystem components that meet the functional and performance subsystem requirements. Physical components include the specific hardware, software, persistent data, and operational procedures. The physical architecture is the foundation for the mission-specific design and development.

## VI. Validation Strategy

It is incumbent on the SSWG to validate the CRM to determine how well it accurately represents the domain of interest (which is not the same as establishing its credentials as an OMG Specification), and a CRM validation strategy has been developed based on Reference [13]. Section 2.2.4 provides ten questions useful in determining the extent to which the model being validated is able to satisfy its intended use:

- Is the model's purpose well-defined?
- Is the model's scope sufficient to meet its intended use?
- Is the model complete relative to its scope?
- Is the model well-formed?
- Is the model consistent?
- Is the model understandable?
- Are modeling conventions documented and used consistently?
- Is the model self-documenting?
- Does the model accurately reflect the domain of interest?
- Does the model integrate with other models?

In the case of a model like the CRM, however, it is difficult for the validators to answer the questions without evaluating concrete examples of a mission-specific model developed using the CRM. For that reason, the SSWG has identified a multi-step approach towards the CRM Validation:

- Identify model attributes which are able to be mapped to one or more of the ten questions.
- Select a CubeSat team of four members of the SSWG, each of whom possesses subject matter expertise in space mission analysis and design, MBSE, and/or SysML and each of whom is able to provide substantive assistance to a CubeSat team to evaluate the as-built CRM and provide a preliminary validation based on her/his expertise.
- Provide this (preliminary) validated CRM to several teams for $\alpha$-testing. This testing will involve each team's using the current version of the CRM to develop its own mission-specific CubeSat (which must in turn be validated by the team).
- De-brief each team and use its comments to perform an additional evaluation of the CRM.
- Develop metrics for determining whether or not the CRM meets its intended use.
- Continue to engage additional teams until the CRM exhibits the appropriate degree of confidence, or until four $\alpha$-tests have been conducted with unsatisfactory results and the approach itself can be re-evaluated.

The SSWG has already shared the current version of the CRM with trusted partners, with encouraging results; and it has arranged for the first $\alpha$-test. While this approach is time-consuming, it has the benefit that it involves actual CubeSat teams with team members of varying levels of expertise, and with different missions.

## VII. Next Steps

The next steps include: 1) adding parametrics for power, weight, and cost, 2) validation of the CRM, and 3) continue working with OMG on making the CRM an OMG specification. Three important parameters for any CubeSat development effort are power, weight, and cost. Value properties for these parameters will be added to blocks at the CubeSat, subsystem, and component levels. Parametric elements, such as constraints and constraint parameters, will be developed to provide the capability to roll-up these values from the component-level up to the CubeSat. The SSWG is currently working with one CubeSat team to apply the CRM to their development effort and is seeking others. As discussed above, data gathered from these experiences will be used as part of a validation of the CRM. The SSWG will continue to prepare the CRM for submission to OMG for consideration as an OMG specification. Part of that effort will include demonstrating that the CRM can be exported from CSM as an XMI file and imported into another modeling tool, such as Sparx System's Enterprise Architect, with no impact to the CRM.

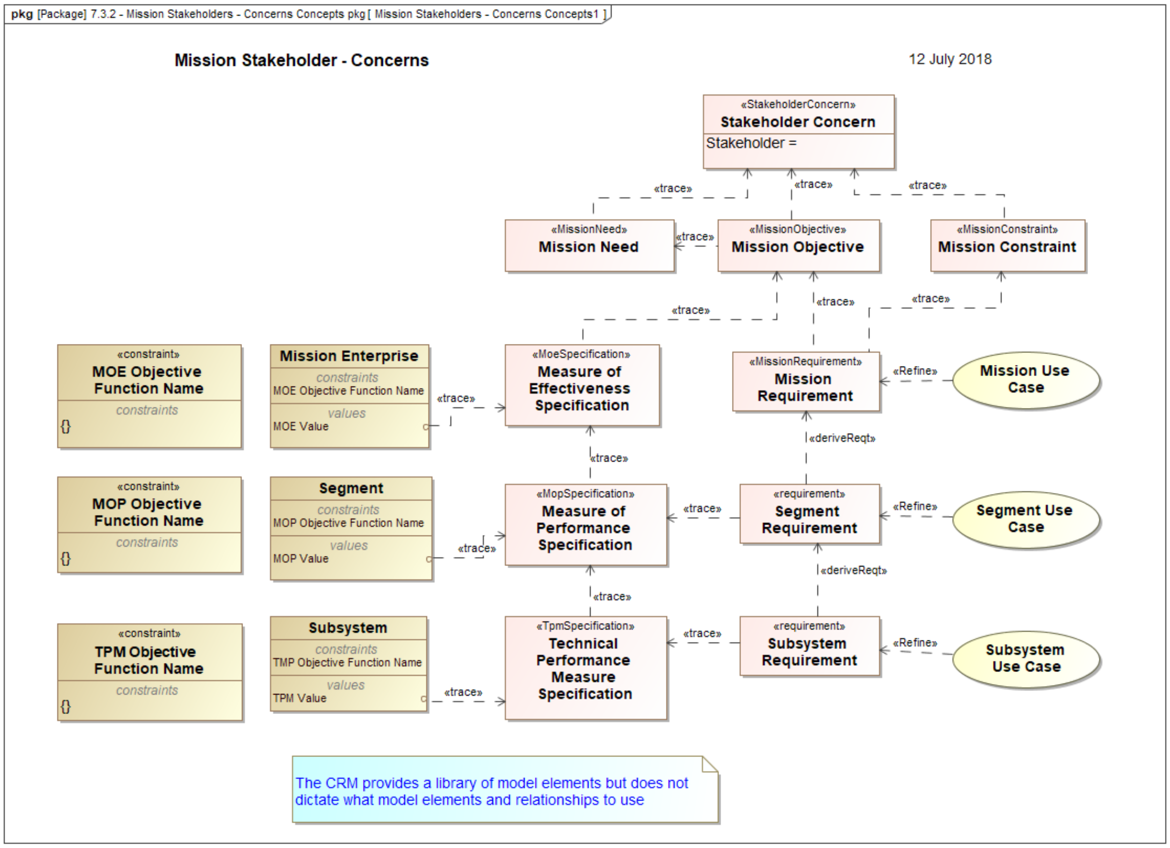



**Fig. 1 Stakeholder Concerns**

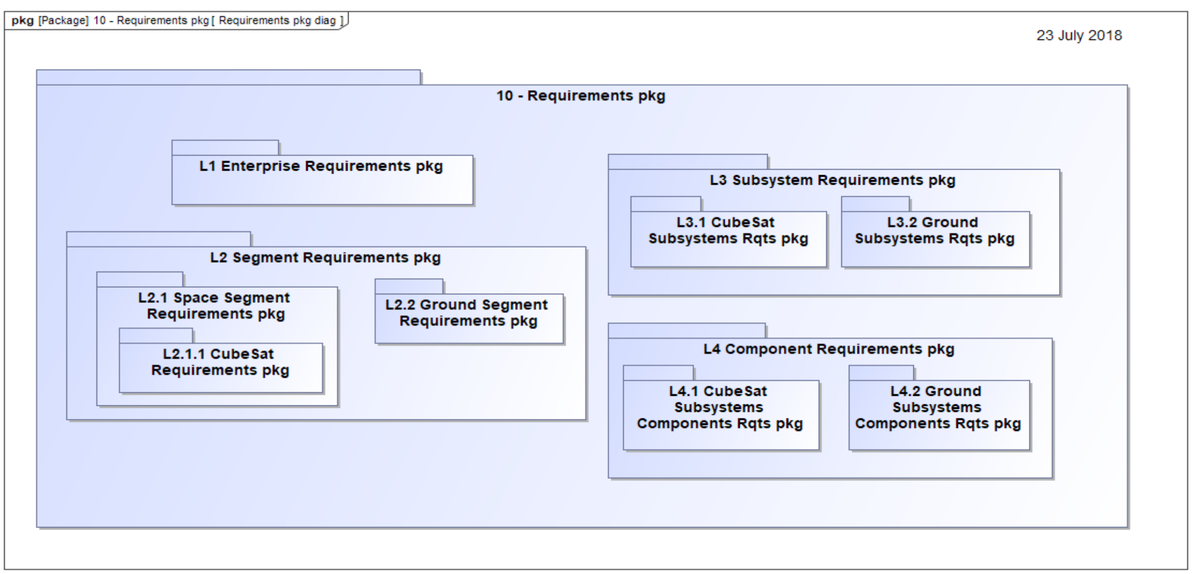



**Fig. 2 Requirements Hierarchy**

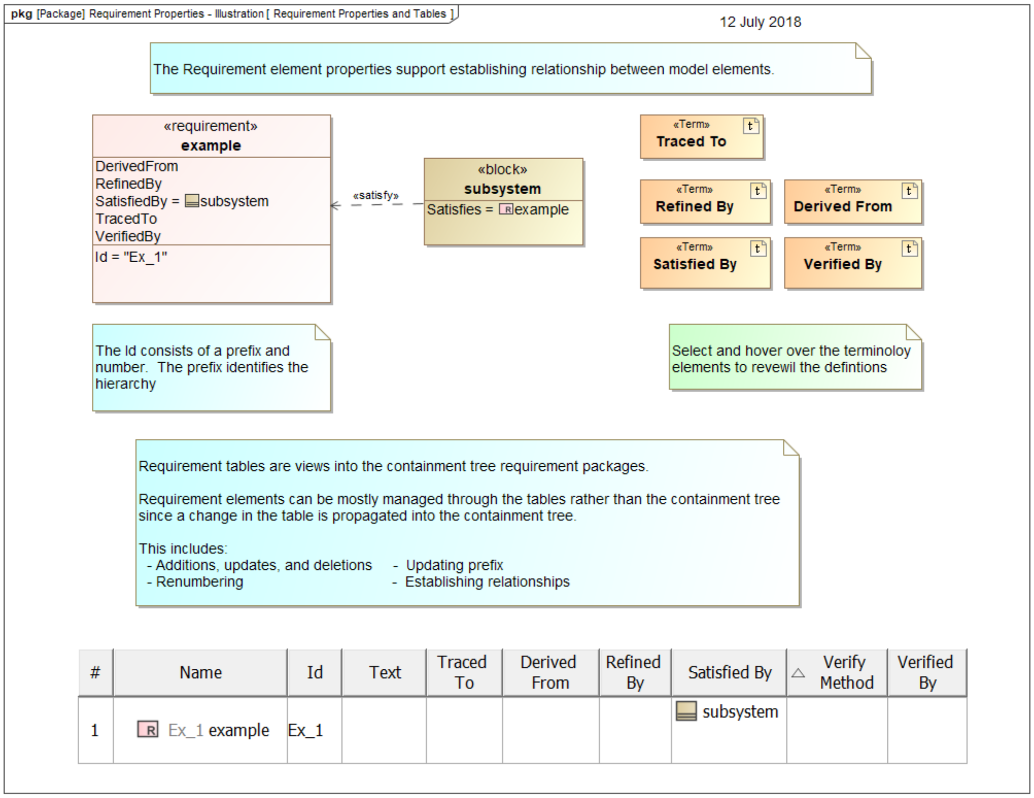



**Fig. 3 Requirements Properties**

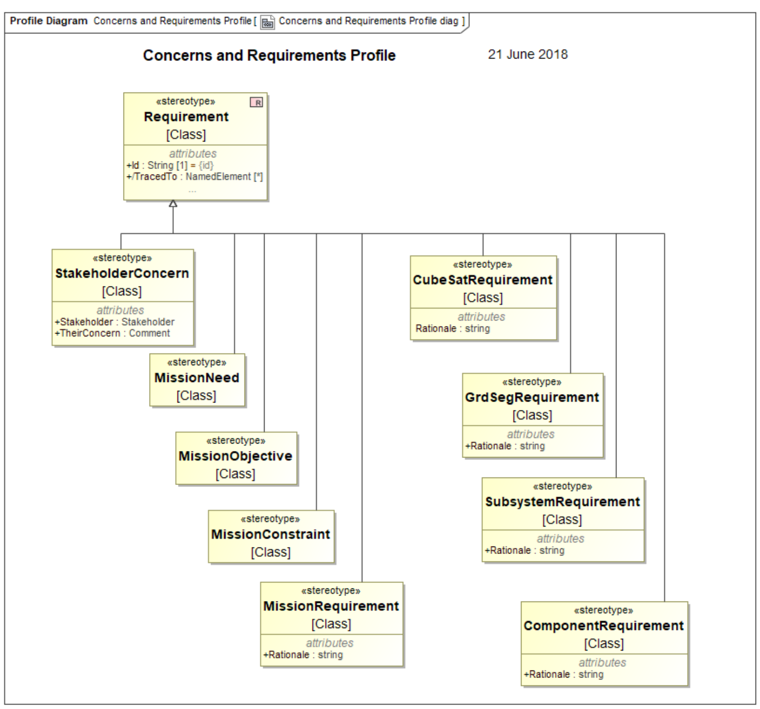



**Fig. 4 Requirement Stereotypes**

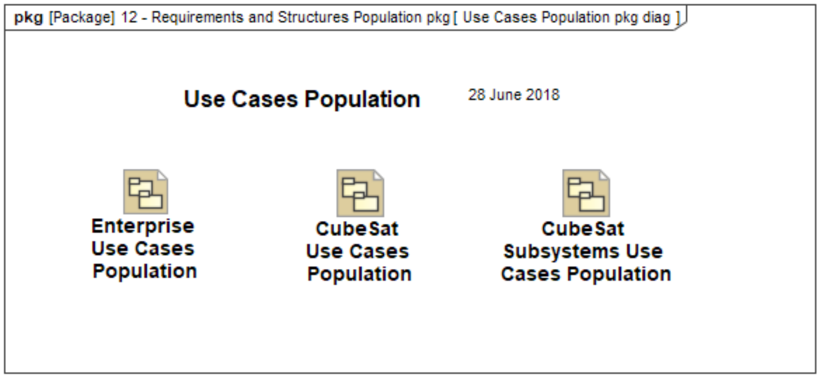
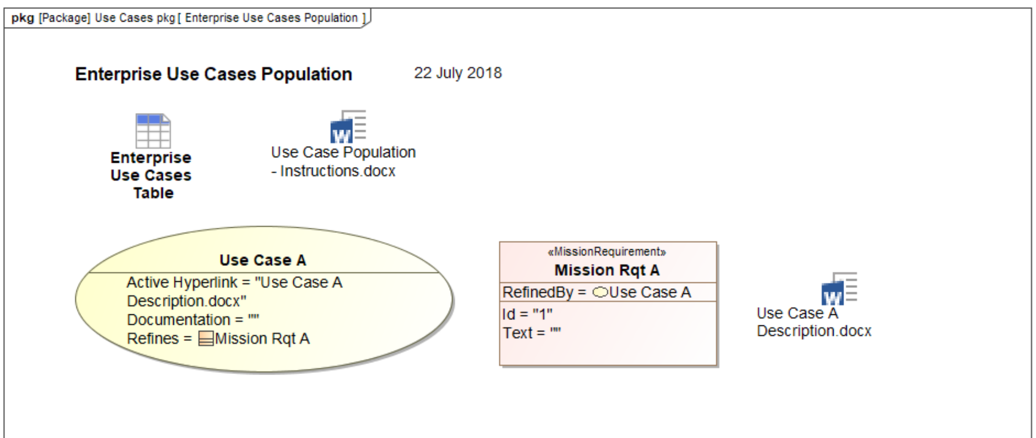



**Fig. 5 Use Cases Population**




| # | Name | Documentation | Refines | Active Hyperlink |
|---|---|---|---|---|
| 1 | Use Case Table Header | | | |
| 2 | Use Case A | | 1 Mission Req A | Use Case A Description.docx |

**Fig. 6 Enterprise Use Cases Population**

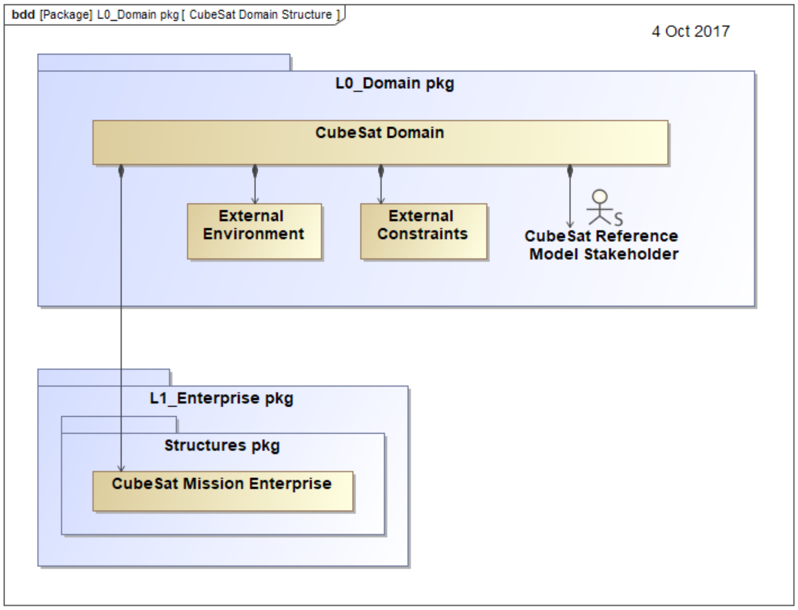



**Fig. 7 CubeSat Domain**

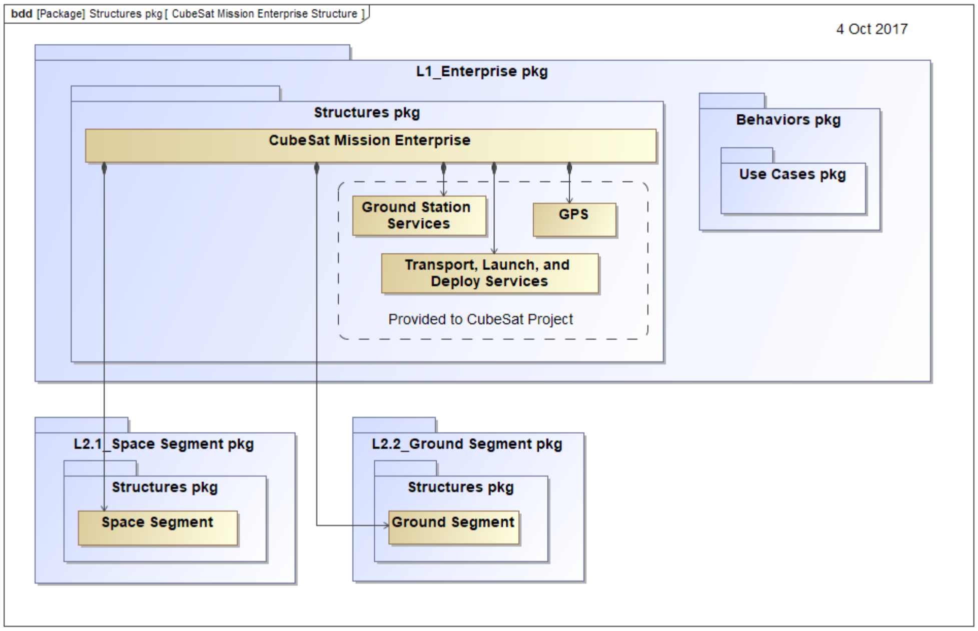



**Fig. 8 CubeSat Enterprise**

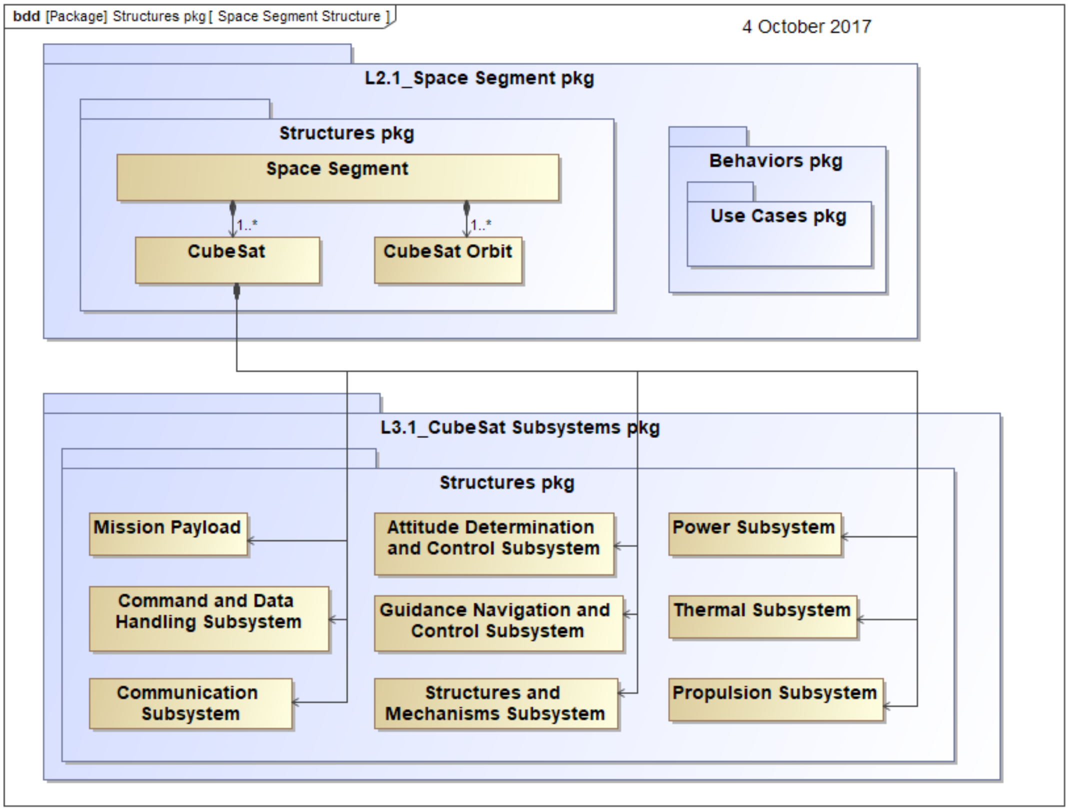



**Fig. 9 CubeSat Subsystems**

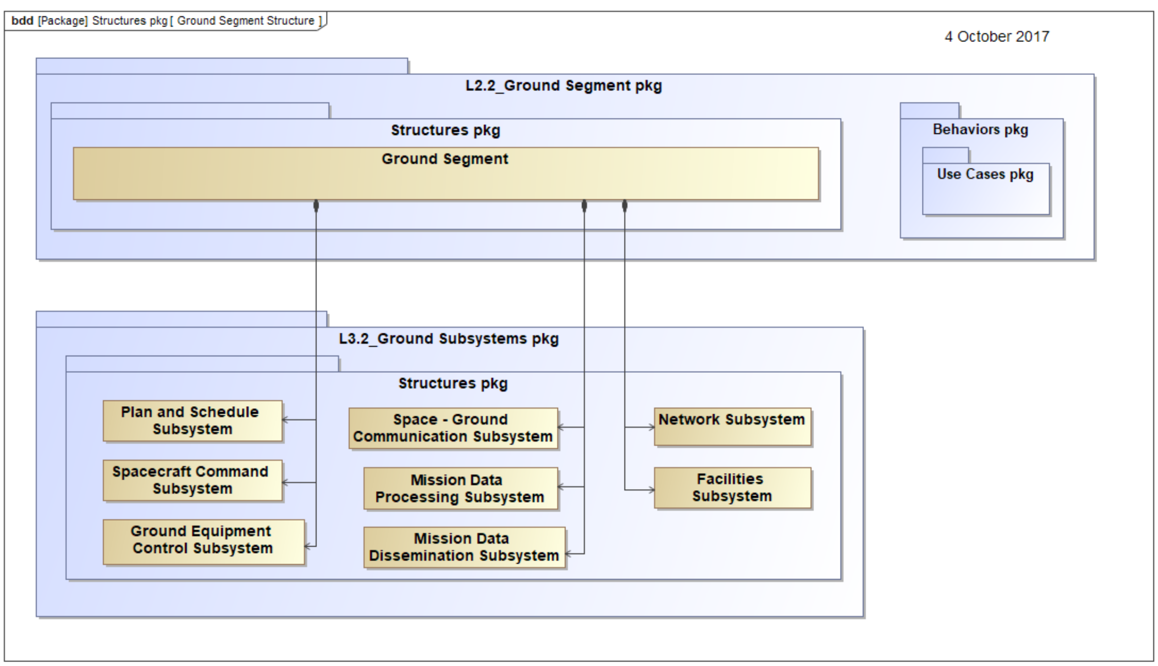



**Fig. 10 Ground Subsystems**

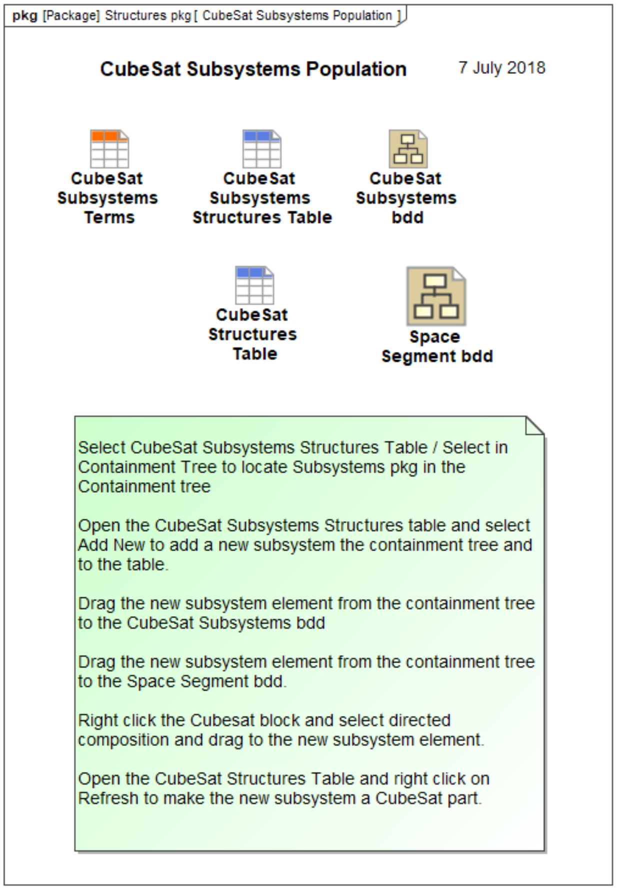



**Fig. 11 CubeSat Subsystems Population**

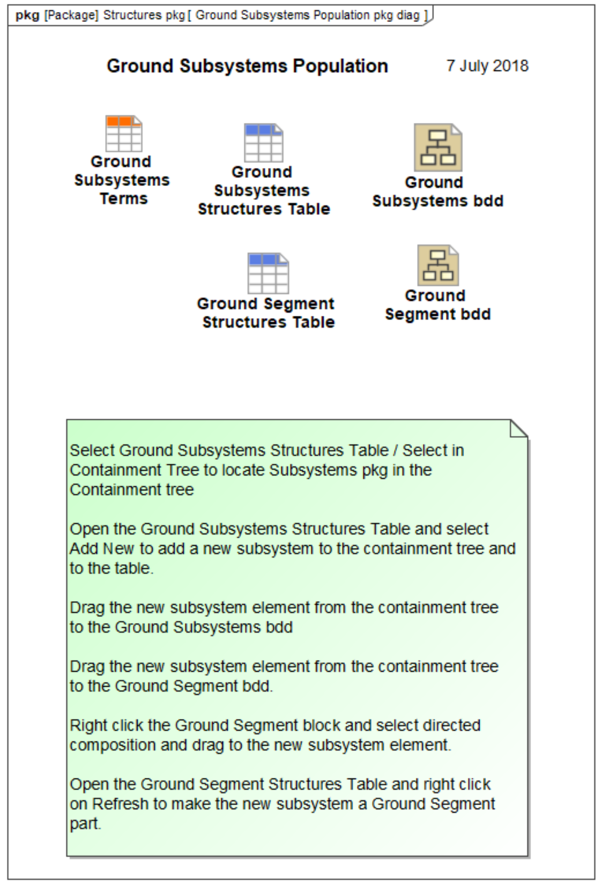



**Fig. 12 Ground Subsystems Population**

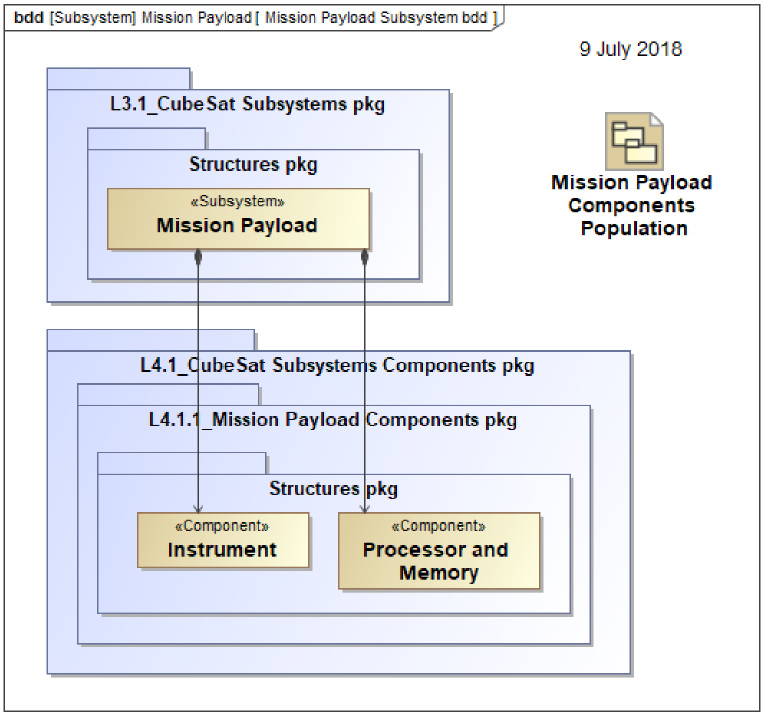



**Fig. 13 CubeSat Mission Payload Representative Components**

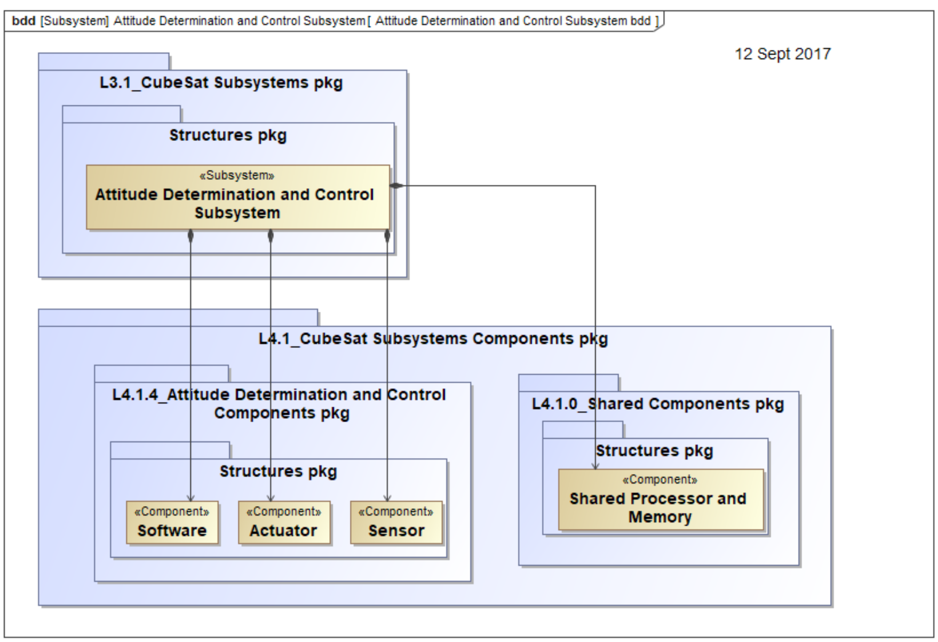



**Fig. 14 CubeSat Attitude Determination and Control Representative Components**

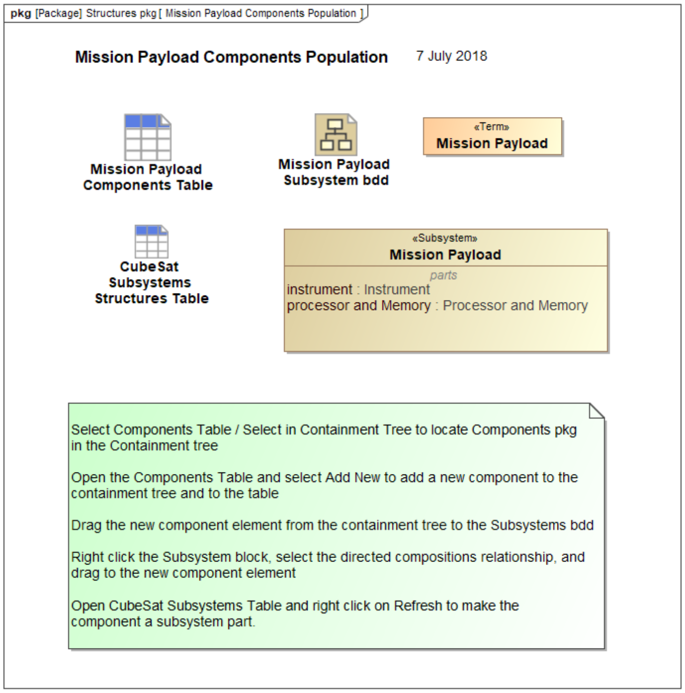



**Fig. 15 CubeSat Mission Payload Components Population**

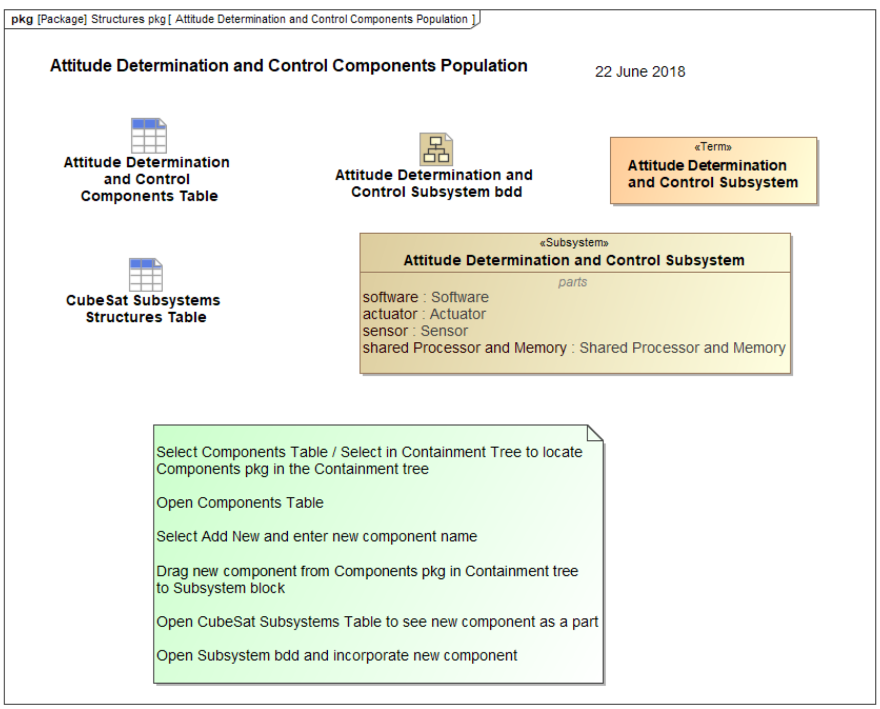



**Fig. 16 CubeSat Attitude Determination and Control Subsystem Components Population**

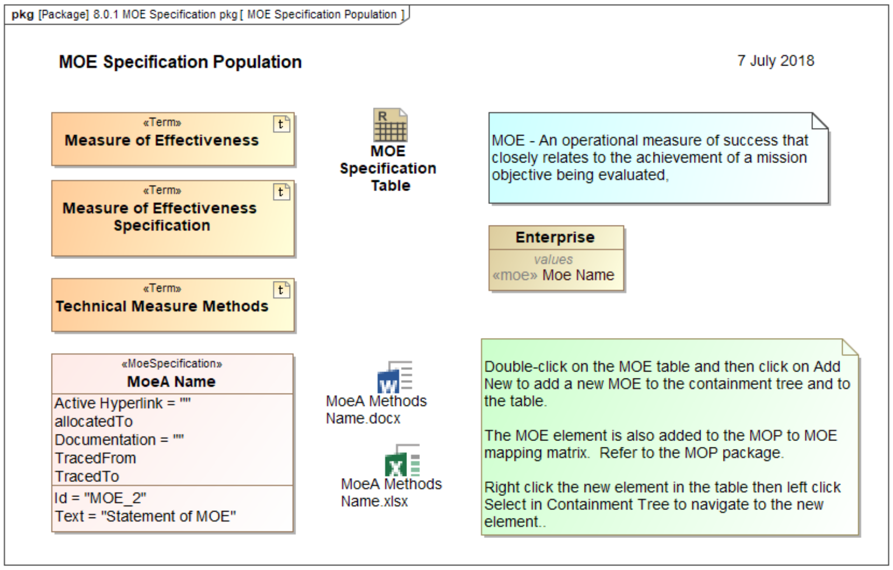



**Fig. 17 MOE Specification Population**

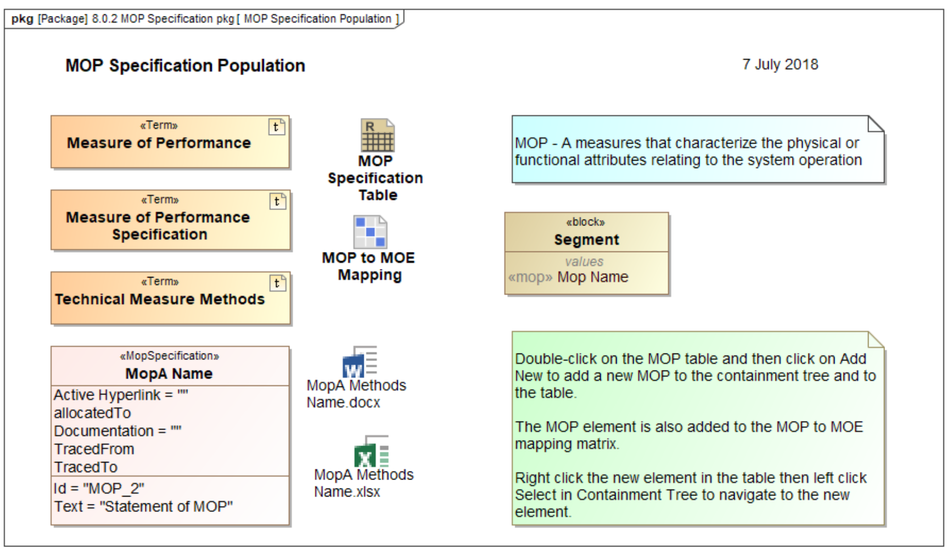



**Fig. 18 MOP Specification Population**

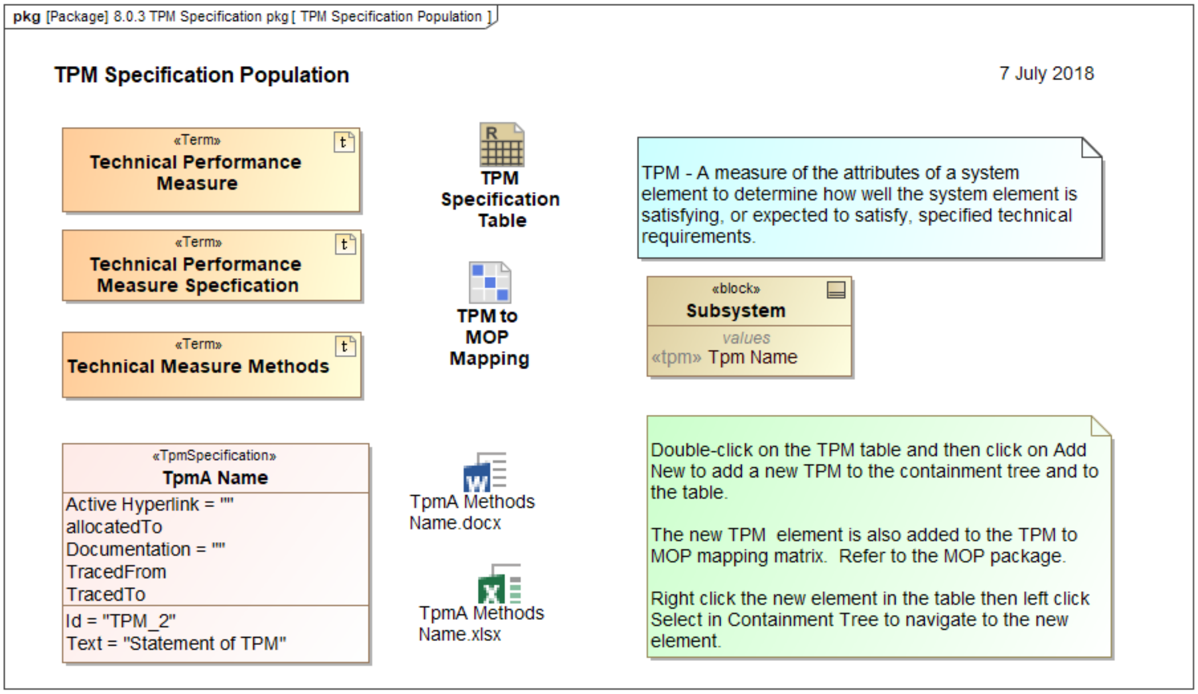



**Fig. 19 TPM Specification Population**

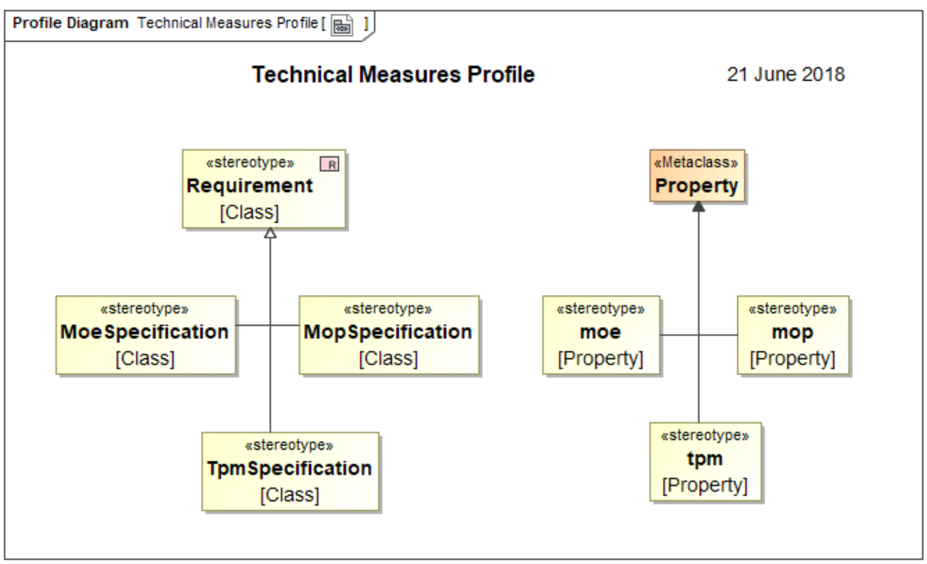



**Fig. 20 Technical Measures Stereotypes**

## References

[1] M. Swartwout, "Attack of the CubeSats: A Statistical Look," *25th Annual AIAA/USU Conference on Small Satellites*. Logan, UT. 2011.

[2] M. Swartwout, "University-Class Spacecraft by the Numbers: Success, Failure, Debris. (But Mostly Success.)," *30th Annual AIAA/USU Conference on Small Satellites*, Logan, UT. 2016.

[3] S. Spangelo, J. Cutler, L. Anderson, E. Fosse, L. Cheng, R. Yntema, M. Bajaj, C. Delp, B. Cole, G. Soremekum, and D. Kaslow, "Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT. 2013.

[4] S. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. Gilbert, L. Hartman, T. Kahn, and J. Cutler, "Applying Model Based Systems Engineering (MBSE) to a Standard CubeSat," in *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2012.

[5] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, "Integrated Model-Based Systems Engineering (MBSE) Applied to the Simulation of a CubeSat Mission," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2014.

[6] D. Kaslow, L. Anderson, S. Asundi, B. Ayres, C. Iwata, B. Shiotani, and R. Thompson, "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model – Interim Status," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2015.

[7] D. Kaslow, B. Ayres, M. Chonoles, S. Gasster, L. Hart, C. Massa, R. Yntema, and B. Shiotani, "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model – Interim Status #2," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, March 2016.

[8] D. Kaslow, B. Ayres, P. Cahill, L. Hart, and R. Yntema, "Developing a CubeSat Model-Based Systems Engineering (MBSE) Reference Model – Interim Status #3," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT. 2017.

[9] D. Kaslow, B. Ayres, P. Cahill, L. Hart, and R. Yntema, "A Model-Based Systems Engineering (MBSE) Approach for Defining the Behaviors of CubeSats," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT. 2017.

[10] D. Kaslow, B. Ayres, P. Cahill, and L. Hart, "A Model-Based Systems Engineering Approach for Technical Measurement with Application to a CubeSat," *Proceedings of IEEE Aerospace Conference*, Big Sky, MT. 2018.

[11] D. Kaslow and A. Madni, "Validation and Verification of MBSE Compliant CubeSat Reference Model," *Proceedings of 15th Annual Conference on Systems Engineering Research*, 2017.

[12] *INCOSE Systems Engineering Handbook*, 4th ed., INCOSE-TP-2003-002-04 2015.

[13] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML*, 3nd ed., Elsevier, Waltham, MA, 2015.

[14] *NASA Systems Engineering Handbook*, rev. 1, December 2007, NASA/SP-2007-6105 Rev1.

[15] J.R. Wertz, D. Everett, and J. Puschell, Eds., *Space Mission Engineering: The New SMAD*, (Space Technology. Library, Volume 28), Hawthorne, CA, Microcosm Press, 2011.